

修士論文

細胞外記録法における単一電極による  
スパイクソーティング

大阪大学大学院生命機能研究科生命機能専攻  
博士前期課程 2006年度 E0167804 番

池田 孟

指導教授 大澤 五住教授

2007年2月19日

# Abstract

## 0.1 要旨

我々の研究室では、単一電極を用いた細胞外記録法により、視覚神経細胞(以下、ニューロン)からの活動電位(スパイク)を記録している。実験の目的は、個々のニューロンが発火するスパイク列と視覚刺激との関係を定量的に解析し、視覚神経系の仕組みを解明する事である。しかしながら、観測したスパイク列は単一のニューロンだけからのスパイクであるとは限らない。電極の周囲には複数のニューロンが存在する事もあるので、観測されたスパイク列は複数のニューロンのスパイクである可能性があるからである。そこで、我々は個々のニューロンのスパイク列を解析するために、まず、複数のニューロンからのスパイクを含む信号から個々のニューロンのスパイクを選り分ける(スパイクソーティング)必要がある。

多くのスパイクソーティングの手法は、個々のニューロンからのスパイク波形の特徴が個々のニューロンについて特異的であるという事実に基づいてソーティングを実行している。従来のスパイクソーティングの手法には大きく分けて2つある(Lewicki,1998)。1つは、ある特定のスパイクの波形をテンプレートとし、スパイク列の他のスパイク波形と比較し、その波形の違いからスパイクソーティングを実行するテンプレート・マッチングと呼ばれる手法である。もう一つは、スパイクの振幅や幅といったスパイクの特徴を表す特徴量をスパイク列の個々のスパイクについて計算し、その特徴量の分布に基づきスパイクをクラスタリングする手法である。しかし実際のスパイク波形のデータは、ノイズの存在や複数のスパイク波形の重複、またバースト発火時におけるスパイク波形の振幅の減衰といった要因で、その波形の特徴が変化してしまうことがある。

最近では、これらスパイク波形に基づかない手法も開発されている(Pouzat, 2004)。Pouzatはスパイクの発火の確率モデルを作り、ベイズ統計を用いて、確率モデルに基づいてスパイクソーティングを実行する手法を開発した。

本研究では、新たにスパイクソーティングの手法を開発した。今回は、離散ウェーブレット変換を用いてスパイク波形の特徴を抽出し、波形の特徴を判別する手法を開発した。またバースト発火時におけるスパイク電位の減衰のモデルからスパイクの発火の確率モデルを作り、その確率モデルに基づくスパイクソーティングの手法も開発した。これら2つの手法を補完する事により、より精度の高いスパイクソーティングを目指した。また、実際の研究室のスパイク波形データをテンプレートに複数のニューロンからのスパイクの発火するデータを再現するスパイクシミュレータを開発し、今回の手法のパフォーマンスを評価した。

# 目次

0.1	要旨	2
<b>1</b>	<b>導入</b>	<b>1</b>
1.1	本研究の背景	1
1.2	本研究の目的	1
<b>2</b>	<b>方法</b>	<b>1</b>
2.1	実験システム	2
2.2	スパイクデータ	2
2.3	スパイクデータの表記について	2
2.4	スパイク発火シミュレータ	3
2.5	ウェーブレット解析によるスパイクソーティング	5
2.6	マルコフ連鎖モンテカルロ法 (MCMC) によるスパイクソーティング	7
2.7	疑似乱数について	8
<b>3</b>	<b>スパイクソーティングシミュレーション</b>	<b>9</b>
3.1	シミュレーション 1	9
3.2	シミュレーション 2	9
3.3	シミュレーション 3	10
3.4	シミュレーション 4	11
3.5	シミュレーション 5	11
<b>4</b>	<b>結果</b>	<b>12</b>
4.1	結果 シミュレーション 1	12
4.2	結果 シミュレーション 2	13
4.3	結果 シミュレーション 3	13
4.4	結果 シミュレーション 4	13
4.5	結果 シミュレーション 5	13
<b>5</b>	<b>考察</b>	<b>14</b>
5.1	ウェーブレット解析によるスパイクソーティングについて	14
5.2	マルコフ連鎖モンテカルロ法によるスパイクソーティングについて	14
<b>6</b>	<b>付録</b>	<b>15</b>
6.1	ウェーブレット解析について	15
6.2	k-means 法	17
6.3	ベイズ統計とマルコフ連鎖モンテカルロ法	17

# 1 導入

## 1.1 本研究の背景

従来のスパイクソーティングの手法には大きく分けて2つある (Lewicki,1998)<sup>7)</sup>。1つは、ある特定のスパイクの波形をテンプレートとし、スパイク列の他のスパイク波形と比較し、その波形の違いからスパイクソーティングを実行するテンプレート・マッチングと呼ばれる手法。もう一つは、スパイクの振幅や幅といったスパイクの特徴を表す特徴量をスパイク列の個々のスパイクについて計算し、その特徴量に基づきスパイクをクラスタリングする手法である。最近のスパイクソーティングの手法には、様々な数理的手法が用いられる事が多くなってきた。離散ウェーブレット変換を用いてスパイク波形の特徴を抽出する手法 (Hullata 2004),<sup>3)</sup> (Quiroga 2004)<sup>12)</sup>。PCA 主成分分析でスパイクの波形を抽出し、reversible-jump MCMC 法で複数のモデルから最適なモデルを選択する手法 (Nguyen 2003)<sup>8)</sup> 等がある。これらスパイク波形に基づかない手法も開発されている (Pouzat,2004)<sup>10)</sup>。Pouzat はスパイクの発火の確率モデルを作り、ベイズ統計を用いて、確率モデルに基づいてスパイクソーティングを実行する手法を開発した。単一電極によるスパイクソーティングだけでなく、ステレオトロード電極やテトロード電極と言った多点電極を用いたスパイクソーティングの手法も盛んであり、ここに紹介した多くのスパイクソーティングの手法も多くは多点電極を対象にした手法である。単純にこれら多点電極を対象にした手法は、単一電極に応用できるものではない。

## 1.2 本研究の目的

本研究の目的は、単一電極でスパイクソーティングを実行する手法を開発する事である。今回は、(Hullata 2004),<sup>3)</sup> (Quiroga 2004)<sup>12)</sup> らの手法を参考に、離散ウェーブレット変換を用いてスパイク波形の特徴を抽出し、波形の特徴を判別する手法を開発した。また、(Pouzat,2002,2004)<sup>10), 11)</sup> を参考に、バースト発火時におけるスパイク電位の減衰のモデルからスパイクの発火の確率モデルを作り、その確率モデルに基づくスパイクソーティングの手法も開発した。本研究では、これら2つの手法を補完する事により、単一電極でも、より精度の高いスパイクソーティング行える手法の開発を目的とした。

# 2 方法

現在の研究室のスパイクの解析は実験中の実験者によるスパイクソーティングの結果を基に行っている。本研究では、実験後に実験で得たスパイクデータを基に (オフライン) スパイクソーティングを行う手法を開発した。実際にスパイクソーティングが必要となる複数のニューロンからのスパイクの発火を再現したスパイクシミュレータを開発し、アルゴリズムのパフォーマンスを調べた。本研究に必要なプログラムは gcc,R 言語で開発した。使用した計算機については、Table 1にまとめた。

## 2.1 実験システム

本研究室の実験システムを Fig. 1に示す。実験は、被験動物に視覚刺激を提示し、神経細胞（以下、ニューロン）の活動電位を単一電極で記録して行われる。単一電極で記録された活動電位はアンプで約 1000 万倍に増幅し、300~5000Hz のバンドパスフィルタをかけられる。その後活動電位は、AD 変換器で 25kHz でサンプリングし、12bit に量子化された離散データに変換される。この時、スパイク判別用 PC (SPCODER) で離散データからスパイク部分が検出される。本研究室の実験システムでは、この SPCODER を使って、実験中に実験者が活動電位に閾値を設定し、活動電位からスパイクを検出し、実験者自信がスパイク波形の特徴を自分の目で判別して、半分手作業にてスパイクソーティング（オンライン・スパイクソーティング）を行っている（森屋 2004）<sup>16)</sup>。スパイクソーティングの結果と各スパイクの波形データ、各スパイクの発火時間は EV ファイル (\*.ev) というバイナリファイルとして記録され、それぞれ実験の種類による名前がつけられている。

## 2.2 スパイクデータ

研究室で保存されているスパイクのデータは、ニューロンの活動電位における各スパイクの波形データとそのときのスパイクの発火時間である。一つのスパイク波形のデータは 40 点のサンプルデータで保存される。実際の研究室で記録したスパイクのスパイク波形の例を Fig. 9に示す。横軸が時間を表すサンプルポイントである。サンプリング周波数が 25kHz であるので、1 点が 0.00004(s)。スパイク波形 1 つは、40 点なので 0.0016(s) となる。縦軸はスパイクの電位を表している。細胞外記録法では、この値はニューロンから離れた単一電極で測定した活動電位をアンプで増幅したものなので、実際の神経細胞のスパイクの電位の値ではない。しかしながら、この値は実際のスパイクの電位を反映したものと言える。

## 2.3 スパイクデータの表記について

以降からスパイクデータを次のようにする。

（1）実際に実験で得られたデータのように、ニューロンの数がいくつか、スパイクがどのニューロンから発火したものが区別できない場合。

- スパイクの数  $n$
- 順に発火したスパイクの発火時間  $t_0, t_1, \dots, t_n$ (秒)
- 順に発火したスパイクのスパイク間隔時間 (ISI)  $\Delta t_1, \Delta t_2, \dots, \Delta t_n$ (秒)
- 順に発火したスパイクの波形データ  $S_1, S_2, \dots, S_n$
- 各スパイクのラベル  $C = \{C_1, C_2, \dots, C_n\}$

実験で  $n$  個のスパイク列を観測したとする。

観測したスパイクについて、その 1 発目に発火したスパイクの発火時間を  $t_0$  で表し、 $k$  発目のスパイクの発火時間を  $t_k$  とする。観測したスパイクの発火時間については全体とし

て、 $t_0, t_1, \dots, t_{n-1}$  とする。各スパイクのスパイク間隔時間 (inter-spike interval time, 以下、ISI) も定義しておく、ISI は、特定のニューロンから発火した連続したスパイクに関して、その隣あったスパイクの発火時間の時間差である。

$$\Delta t_{k-1} = t_k - t_{k-1} \quad (2.1)$$

と定義される。実験では、スパイクデータがスパイクの波形データについては、波形データが 40 個のサンプル点で表されているので、一つのスパイク波形データを 40 次元のベクトルで表す。k 発目に発火したスパイクのスパイク波形データを  $\mathbf{S}_k$  とすると、 $\mathbf{S}_k = (s_0, s_1, \dots, s_{39})$  となる。観測したスパイク波形のデータは、発火した順に  $\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_n$  とする。仮にこのスパイク列が k 個のニューロンから発火したものとする。i 番目のスパイクに対し、 $C_i = X (X = 1, 2, \dots, n)$  とスパイクを発火したニューロンの番号を割り当てる。実際には、観測したスパイクデータでは、各スパイクがどのニューロンが発火したスパイクかは分からない。スパイクソーティングでは、i 番目のスパイクのラベル  $C_i$  に、アルゴリズムによって番号を割り当てる。

(2) スパイク発火シミュレータ (後述) で作ったスパイクデータのように、スパイクがどのニューロンから発火したものか区別がつく場合。スパイク列が k 個のニューロンから発火したものとし、

- ニューロンの番号  $i, (i = 1, 2, 3 \dots k)$
- スパイクの数  $n$ , ニューロン  $i$  が発火したスパイクの数  $n_i$
- ニューロン  $i$  が発火したスパイクの数  $n_i$
- ニューロン  $i$  が順に発火したスパイクの発火時間データ  $t_0^{(i)}, t_1^{(i)}, \dots, t_{n_i}^{(i)}$  (秒)
- ニューロン  $i$  が順に発火したスパイクのスパイク間隔時間 (ISI)  $\Delta t_1^{(i)}, \Delta t_2^{(i)}, \dots, \Delta t_{n_i}^{(i)}$  (秒)
- ニューロン  $i$  が順に発火したスパイクのの波形データ  $\mathbf{S}_1^{(i)}, \mathbf{S}_2^{(i)}, \dots, \mathbf{S}_{n_i}^{(i)}$

i 番目のニューロンについて、その 1 発目に発火したスパイクの発火時間を  $t_0^{(i)}$  で表し、k 発目のスパイクの発火時間を  $t_k^{(i)}$  とする。i 番目のニューロンが発火したスパイクの発火時間については全体として、 $t_0^{(i)}, t_1^{(i)}, \dots$  とする。i 番目のニューロンから発火されたスパイク波形のデータも、発火した順に  $\mathbf{S}_0^{(i)}, \mathbf{S}_1^{(i)}, \dots$  とする。

## 2.4 スパイク発火シミュレータ

研究室の実験で得られたスパイクデータを再現するために、スパイクデータのシミュレータを作成した。シミュレータでは現実の実験データに近づけるため、実際の研究室で観測したスパイクの波形データをテンプレートとして、次の事を再現した。

- 複数のニューロンによるスパイクの発火列。
- 観測されるスパイク波形データに付加するノイズ。

- スパイク電位のバースト発火による減衰。
- 2つのスパイクの発火が実験データでの40点(0.0016s)以内に起きたときにおこる、重複したスパイクの波形データ。

シミュレータはテンプレート波形を設定した発火時間順に時間軸上に並べ、その後にテンプレート波形に条件を設けて、スパイク波形を作る。シミュレータは以下のような手順でシミュレーションデータを生成する。

(1) スパイク波形のテンプレートの用意。

まず、異なる複数のニューロン $i$ 個（ニューロンの番号を1から $i$ 番とする）から発火した $i$ 個のスパイク波形データを用意する。それぞれの波形データをテンプレートとし、それを波形データと同じく **template<sup>(i)</sup>** と40次元のベクトルで表す。波形データは、異なるEVファイルから、スパイク波形を読み込むことで用意する。

(2) 発火時間の設定。

それぞれのニューロンからのスパイクの発火時間を設定する。発火時間にはポアソン過程を仮定する。R言語の関数 `rpois()` を使用して、平均値  $\lambda_{Pois}$  のポアソン分布から疑似乱数を生成する。

$$\Delta t_j^{(i)} \sim Pois(\lambda_{Pois}) \quad (2.2)$$

これで全てのスパイクのISIが定義できたので、テンプレートを時間軸上に並べる事ができる

(3) スパイク波形データの減衰。

スパイク波形をISIに基づいて減衰させる。

$$\mathbf{s} = (1 - \delta \exp(-\lambda \Delta t_i^{(i)})) \mathbf{template}^{(i)} \quad (2.3)$$

$\delta, \lambda$  は定数である。上式に基づいてスパイク電位を Fig. 5 のように ISI が短ければ短いほど大きく減衰させる。スパイクを一つ生成するごとに、 $\lambda$  にノイズを加えて減衰率を変化させる。上式については、(Fee 1996)<sup>5)</sup> で提案されている。本研究室でも(吉川 2004)<sup>14)</sup> において、バースト発火時において、スパイク電位が減衰することが確認された。

また、 $\lambda$  にスパイクの発火の度にノイズを付加して、減衰率が発火ごとに変動させる。ノイズは正規分布  $N(0, \tau^2)$  から生成して、 $\lambda$  に付加する。

$$\lambda = \lambda + \epsilon \quad (2.4)$$

$$\epsilon \sim N(0, \tau_1^2) \quad (2.5)$$

(4) スパイク波形データへのノイズの付加。

ここで、スパイク波形にノイズを加える。ノイズは正規分布から疑似乱数をスパイク波形のデータ数 (40 個) を生成する。

$$\mathbf{noise} = n_1, n_2, \dots, n_{39} \sim N(\mu, \tau_2^2) \quad (2.6)$$

$$\mathbf{S}_i \leftarrow \mathbf{S}_i + \mathbf{noise} \quad (2.7)$$

(5) スパイク波形同士の重複の再現。

このように各スパイクの発火時間とスパイク波形データを作った後に、各スパイクの発火時間をチェックし、もし 2 つのスパイク時間の間隔がスパイク波形データ一つ分 (0.0016s) 以内にであったときには、スパイクが重複したものとして重複したスパイク波形を作成し、スパイクを重複したものに置き換える。

仮に、2 つのスパイク波形  $\mathbf{S}_a, \mathbf{S}_b$  が重複したとすると、 $\mathbf{S}_a = (a_1, a_2, \dots, a_{40})$ ,  $\mathbf{S}_b = (b_1, b_2, \dots, b_{40})$  のそれぞれの波形の重複した後の波形 **overlapped**  $\mathbf{S}_a = (a'_1, a'_2, \dots, a'_{40})$ , **overlapped**  $\mathbf{S}_b = (b'_1, b'_2, \dots, b'_{40})$  は、

$$a'_i = \max(a_i, b_i) \quad (2.8)$$

$$b'_i = \max(a_i, b_i) \quad (2.9)$$

$$i = 1 \sim 40 \quad (2.10)$$

として、重複が生じた区間 (40 点) におけるデータにおいて、 $\mathbf{S}_1, \mathbf{S}_2$  の大きい方のスパイク波形のデータを重複したスパイク波形として残すようにする。

## 2.5 ウェーブレット解析によるスパイクソーティング

ウェーブレット解析によるスパイクソーティングは、スパイクの特徴を抽出し、スパイク波形の判別をするという 2 つのステップからなる。Fig. 2 にスパイクソーティングのイメージ図を載せた。一段目 (Spikes) は 3 種類のスパイク波形のテンプレートである。二段目 (Feature extraction) は波形の特徴を抽出できる特徴空間の作成。三段目 (k-means clustering) は特徴空間でのクラスタリング (判別) である。

ウェーブレット解析によるスパイクソーティングの手順は次のようになる。

- (1) 各スパイク波形データをウェーブレット変換する。(Feature extraction)
- (2) クラスタリングに適した基底関数を選ぶ。(Feature extraction)
- (3) 選んだ基底関数のから作られた特徴空間上のスパイク列をクラスタリングを行う。(k-means clustering)

各手順について詳しく記す。



(1) スパイク波形データのウェーブレット変換。

各スパイク波形データ  $\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_n$  をそれぞれ 3 次のスプライン関数で補完して各スパイク波形データを 40 点のデータから 64 点のデータ  $\mathbf{S}'_0, \mathbf{S}'_1, \mathbf{S}'_n$  にする。このスプライン補完は、多重解像度解析では、 $2^x$  のデータ点が必要だからである。

次に、レベル 8 の多重解像度解析による 64 点で補完した各スパイク波形データ  $\mathbf{S}'_0, \mathbf{S}'_1, \mathbf{S}'_n$  をウェーブレット変換する。各スパイク波形のデータを  $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1, \dots\}$  と表す。ウェーブレット関数には、 $N=8$  の Daubechies のウェーブレット関数 Fig. ?? を選んだ。ウェーブレット変換されたスパイク波形は  $\mathbf{w}_{(i)} = \{w_0, w_1, \dots, w_{63}\}$  64 次元のデータに変換される。ここで、 $w_0, w_1, \dots, w_{63}$  は、ウェーブレットの基底関数の各係数である。つまり、一つのスパイク波形  $\mathbf{S}'$  は、ウェーブレットの基底関数  $\phi_1 \sim \phi_{64}$  を用いて、

$$\mathbf{S}' = \sum_{i=0}^{63} w_i \phi_i \quad (2.11)$$

と表される。

(2) 基底関数の選択。

64 個のウェーブレット変換の基底関数から、クラスタリングに適した基底関数を選ぶ。各基底関数の得点  $score_i, i = 1 \sim 64$  を次のように各係数の標準偏差で定義し、この得点が高ければ高いほど基底関数をクラスタリングに適した基底関数と考える。

$$score_i = \frac{1}{n-1} \sum_{j=1}^n (w_i - \bar{w})^2 \quad (2.12)$$

$$\bar{w} = \frac{1}{n} \sum_{j=1}^n w_i \quad (2.13)$$

得点を基底関数の係数の標準偏差で定義したのは、標準偏差が大きい基底関数ならば特徴空間でのスパイクの分布が広く、クラスタリングの際に複数のクラスターができていく可能性が大きいだろうと考えたからである。この得点の高い上位 10 個の基底関数で特徴空間を構成する。

(3) クラスタリング。

10 次元の特徴空間上のスパイク列を k-means 法でクラスタリングする。k-means 法とは、最も簡単なクラスタリングの手法の一つである。与えられた多次元のデータを、クラスターの平均を計算して、k 個のクラスターに判別する。k-means 法では、特徴空間のスパイク列を何個のクラスターに分割するか k の値をユーザが入力しなければならない。本研究では、スパイク発火シミュレータで設定したニューロンの個数と同じ個数にクラスターを設定して、特徴空間のスパイク列を分割する。実際の実験データを本研究の手法でスパイクソーティングする場合は、スパイク列の波形や特徴空間上のスパイク列を観察して、ユーザがクラスターの数を設定する事になる。その意味では、本研究の手法は semi-automatic な手法と言える。

ウェーブレット解析と k-means 法については、(付録 6.1, 6.2) にまとめておいた。

## 2.6 マルコフ連鎖モンテカルロ法 (MCMC) によるスパイクソーティング

この手法は、スパイク電位の減衰にを考慮に入れたスパイクソーティングの手法である。この手法では、スパイク電位の減衰に関する確率モデル式 (2.15) を用いて、ソーティングを行う。最初に、初期値として、正しいかどうかはともかく、各スパイクに適当なスパイクラベルを設定しておく。この仮定したスパイクラベルを確率モデルと比較して、修正する（正しくないと考えれば、ラベルを他のラベルに変更する）というステップを反復する事により、正しいラベルに近づけるというソーティングの手法である。このような手法は、Pouzat によって最初に開発された (Pouzat 2004),<sup>2)9)</sup> ただし、Pouzat の確率モデルは、スパイクの電位の減衰だけでなく、ISI Histogram の確率分布も考慮に入れている。また、多点電極を対象にしたものである。本研究との違いは、本研究の確率モデルはスパイク電位の減衰のみを確率モデルに考慮し、この手法をウェーブレット解析によるスパイクソーティングの補完に使用しようとする点である。

この手法は、ベイズの定理とマルコフ連鎖モンテカルロ法 (MCMC) によって実装された。ベイズの定理とマルコフ連鎖モンテカルロ法 (MCMC) については、付録に書いた。

### 2.6.1 減衰の確率モデル

スパイク電位の減衰モデル Fig. 5 a はスパイクの電位を表し（スパイク波形のピーク値や波形のデータ）、 $a_{pre}$  はひとつ手前のスパイクの電位を表す。 $\delta$ 、 $\lambda$  は定数である。

$$a = (1 - \delta \exp(-\lambda \Delta t)) a_{pre} \quad (2.14)$$

このモデルでは、ISI が短ければ短いほど減衰率が大きくなる。上式については、(Fee 1996)<sup>5)</sup> で提案されている (Fig. 7)。

本研究室でも (吉川 2004)<sup>14)</sup> において、バースト発火時において、スパイク電位が減衰することが確認された。

式 (2.14) を基に、次のような確率モデルを作った。確率モデルでは、モデルから予測される電位式 (2.14) と実際の電位との誤差が  $a - a_{pre}(1 - \delta \exp(-\lambda \Delta t))$  である確率の確率分布を正規分布  $N(0, \tau^2)$  であるとした。Fig. 7

$$P_{amp}(a|\Delta t, \delta, \lambda) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2} |a - a_{pre}(1 - \delta \exp(-\lambda \Delta t))|^2\right) \quad (2.15)$$

$$\sigma = 0.1 (1 - \delta \exp(-\lambda \Delta t)) a_{pre} \quad (2.16)$$

ここで、標準偏差  $\sigma$  は、モデルから予測される電位式 (2.14) の 0.1 倍とした。

$n$  個のスパイク電位の値と発火時間の分かるスパイク列を  $D = \{(a_0, \Delta t_i), (a_0, \Delta t_i), \dots, (a_n, \Delta t_n)\}$  と表す。確率モデル式 (2.15) から、スパイク列  $D$  を観測する確率を考える。最初に、一つのニューロンが連続して  $n$  個のスパイクが発火している状況を考える。2つの前後するスパイクの確率モデルは、式 (2.15) で先ほど立てたから、連続して  $n$  個のスパイクが発火して、観測データ  $D$  を観測する確率  $P(D|\delta, \lambda)$  は

$$P(D|\delta, \lambda) = \prod_i P_{amp}(a_i|\Delta t_i, \delta, \lambda) \quad (2.17)$$

となる。

次に、 $i(i = 1, 2, \dots)$  個のニューロンから発火されるスパイク列の発火を考える。各  $i$  個のニューロンについて、式 (2.17) と同じように考えればよいから、各ニューロン  $i(i = 1, 2, \dots)$  の  $\delta, \lambda$  を  $\delta_i, \lambda_i$  とし、

$$P(D|\delta, \lambda) = \prod_{\text{各ニューロン } i(i=1,2,\dots) \text{ について}} \left( \prod_i P_{amp}(a_i|\Delta_i, \delta_i, \lambda_i) \right) \quad (2.18)$$

現実には、我々は観測データ  $D$  を知る事ができるが、各ニューロン  $i(i = 1, 2, \dots)$  の  $\delta_i, \lambda_i$  については分からない。しかし、 $\delta_i, \lambda_i$  に確率分布を仮定して、事前分布  $P(\delta_i)$ 、 $P(\lambda_i)$  と設定すると、尤度関数は式 (2.18) より  $P(D|\delta_i, \lambda_i)$  だから、

ベイズの定理 (付録 6.3.1) より、その事後確率は、

$$P_{post}(\delta, \lambda|D) = \frac{P(D|\delta, \lambda)P(\delta)P(\lambda)}{Z} \quad (2.19)$$

$$\propto P(D|\delta, \lambda)P(\delta)P(\lambda) \quad (2.20)$$

となり、事前確率  $P(\delta_i)$ 、 $P(\lambda_i)$  と尤度関数  $P(D|\delta_i, \lambda_i)$  から、事後確率  $P_{post}(\delta_i, \lambda_i|D)$  が計算できる。ここで  $Z$  は規格化定数で、 $Z = \int P(D|\delta, \lambda)P(\delta)P(\lambda)d\delta d\lambda$  であるが、メトロポリス-ヘイスティングスアルゴリズム (付録 6.3.3) を用いる際には、 $Z$  を計算する必要は無い。

## 2.6.2 MCMC 法でのスパイクソーティング

事前分布として一様分布  $\delta_i, \lambda_i$  を設定する ( $P(\delta_i) \subset (0.1, 0.9), P(\lambda_i) \subset (1, 200)$ )。  $\delta_i, \lambda_i$  の提案分布として、酔歩過程 (付録 6.3.3.1) を採用した。

メトロポリス-ヘイスティングスアルゴリズム (付録 6.3.3) を用いて、事後分布  $P_{post}(\delta_i, \lambda_i|D)$  を計算する。実際に事後確率を計算する際には、式 (2.20) の両辺に対数をとった、対数尤度関数

$$\log P_{post}(\delta_i, \lambda_i|D) = \log P(D|\delta_i, \lambda_i) + \log P(\delta_i) + \log P(\lambda_i) + C \quad (2.21)$$

を計算する。

スパイク  $n$  個のスパイクラベル  $C_i$  と各ニューロンのパラメータについて

$$C_i^{(t-1)} \rightarrow \delta_{C_i}^{(t-1)} \rightarrow \lambda_{C_i}^{(t-1)} \rightarrow C_{i+1}^{(t-1)} \rightarrow \dots \rightarrow \dots \rightarrow C_i^{(t)} \rightarrow \dots \quad (2.22)$$

とメトロポリス-ヘイスティングスアルゴリズムのステップを繰り返せば、式 (2.21) が収束し、そのときのスパイクラベル  $C_i$  は正しくソーティングされるはずである。

MCMC によるスパイクソーティングは、統計物理における Potts モデルと呼ばれる磁性体のスピンモデルのモンテカルロ・シミュレーションと良く似ている。スピンの向きをスパイクのラベルの番号に、磁性体のエネルギーが式 (2.20) に対応している。

## 2.7 疑似乱数について

スパイクシミュレータや MCMC 法では、大量に疑似乱数を生成して使用する必要がある。本研究では疑似乱数の生成アルゴリズムについては、“Mersenne-Twister 法”を採用した。R 言語では (Matsumoto and Nishimura 1998)<sup>6)</sup> の twisted GFSR 法が標準の疑似乱数生成関数となっている<sup>13)</sup>。この手法は、

周期  $2^{19937-1} (= 4.315 \times 10^{60001})$  で、すべての周期に関し 623 次元空間に均一分布する一様乱数を生成する。これは従来の線形合同法等の手法に比べ格段に長い周期である。乱数種は 32 ビット整数の 624 次元ベクトル+その中の現在位置を示す整数である。

## 3 スパイクソーティングシミュレーション

前章で紹介した2つのスパイクソーティングのパフォーマンスを調べるため、スパイクシミュレータで作成したスパイクデータを使った5つのシミュレーションを行った。

### 3.1 シミュレーション1

3個のニューロンからのスパイク列をスパイクソーティングのパフォーマンスを調べた。各ニューロンのスパイクが減衰による電位の変動が小さいときに、ウェーブレット解析によるスパイクソーティングだけでスパイクソーティングのパフォーマンスがどうなるかを調べた。

#### 3.1.1 スパイクシミュレータの設定

3個のスパイク波形のテンプレートを、実験室で記録したスパイクデータ

- ニューロン 1 04-9A\_ep013.005.ev
- ニューロン 2 03-35Bep008.004.ev
- ニューロン 3 0310d\_ep031.003.ev

から、用意した。それぞれの波形を Fig. 9に描いた。

スパイク発火シミュレータを使って、

- ニューロン 1 35 個  $\delta = 0.8, \lambda = 100$
- ニューロン 2 35 個  $\delta = 0.8, \lambda = 100$
- ニューロン 3 30 個  $\delta = 0.8, \lambda = 150$

と合計 100 発ずつ発火した。それぞれの平均発火率は、 $\lambda_{pois} = 0.15, 0.17, 0.20(s)$  とした。シミュレータの設定値については、表 2にまとめた。

#### 3.1.2 ウェーブレット解析によるスパイクソーティング

各スパイク波形を *level-6* でウェーブレット変換 (多重解像度解析) した。k-means 法でクラスターの個数を 3 個に設定して、スパイクソーティングを行った。

### 3.2 シミュレーション2

3個のニューロンからのスパイク列をスパイクソーティングのパフォーマンスを調べた。各ニューロンのスパイクの中に、減衰による電位の変動が大きいスパイクが存在する時、ウェーブレット解析によるスパイクソーティングだけで、スパイクソーティングのパフォーマンスがどうなるかを調べた。

ニューロン 1、2 を電位の変動が大きい (ISI が短い) ニューロン。ニューロン 3 電位の変動が小さい (ISI が長い) を平均発火率が低いニューロンと設定した。

### 3.2.1 スパイクシミュレータの設定

3 個のスパイク波形のテンプレートを、実験室で記録したスパイクデータ

- ニューロン 1 038c\_\_ep025.006.ev
- ニューロン 2 0320cep036.015.ev
- ニューロン 3 043c\_\_ep012.010.ev

から、用意した。それぞれの波形を Fig. 12 に描いた。

スパイク発火シミュレータを使って、

- ニューロン 1 35 個  $\delta = 1.0, \lambda = 200$
- ニューロン 2 35 個  $\delta = 1.0, \lambda = 150$
- ニューロン 3 30 個  $\delta = 1.0, \lambda = 150$

と合計 100 発ずつ発火した。それぞれの平均発火率は、 $\lambda_{pois} = 0.020, 0.035, 0.200(s)$  とした。

ニューロン 1 とニューロン 2 が短い ISI をもつニューロンで、ニューロン 3 が長い ISI をもつニューロンである。シミュレータの設定値については、表 3 にまとめた。

### 3.2.2 ウェーブレット解析によるスパイクソーティング

各スパイク波形を *level-6* でウェーブレット変換 (多重解像度解析) した。k-means 法でクラスターの個数を 3 個に設定して、スパイクソーティングを行った。

## 3.3 シミュレーション 3

MCMC 法によるスパイクソーティングのパフォーマンスを調べた。まず、MCMC によるスパイクソーティングのパフォーマンスがどの程度あるのかを調べるために、スパイクの初期値をランダムに設定して MCMC 法によるスパイクソーティングのみでどれだけ、どの程度のパフォーマンスを上げられるのか調べた。スパイクの数は、シミュレーション 2 のデータを使って、データの最初の 20 個に限定して調べてみた。

### 3.3.1 スパイクシミュレータの設定

スパイクシミュレータの設定は、3.2 と同じものとした。最初の 20 個のスパイクのデータを使った。各ニューロンのスパイクの数は

- ニューロン 1 12 個
- ニューロン 2 7 個
- ニューロン 3 1 個

となった。

### 3.3.2 MCMC の初期設定

ニューロン 1,2,3 パラメータ  $\delta, \lambda$  の事前分布を一様分布で設定した。 $\delta_1, \delta_2, \delta_3$  の事前分布の定義域を 0.1 から 0.9 とした。また、 $\lambda_1, \lambda_2, \lambda_3$  の事前分布を定義域を 1 から 200 とした。

$$\delta_1, \delta_2, \delta_3 \subset u(0.1, 0.8) \quad (3.1)$$

$$\lambda_1, \lambda_2, \lambda_3 \subset u(1, 200) \quad (3.2)$$

初期値のパラメータの値を  $\delta = 0.5, \lambda = 100.5$  とした。初期値のスパイラベル  $C = \{C_1, C_2, \dots, C_n\}$  をランダムに設定した。各スパイクのラベルの初期値がどのような番号に設定されたのかは、?? に記した。

パラメータの候補分布には、酔歩過程式 (6.20) をもちいて、 $\tau$  の値をそれぞれの定義域の 0.1 倍とした。スパイクラベルの候補には、現在と異なるラベルを等確率で候補として生成した。各スパイクラベルについて 100 回、各パラメータ  $\delta_1, \delta_2, \delta_3, \lambda_1, \lambda_2, \lambda_3$  については各スパイクラベルを調べるごとに 1 回、合計 30000 回、メトロポリス-ヘイスティングスアルゴリズムのステップを反復した。

## 3.4 シミュレーション 4

シミュレーション 2 のスパイクデータを使って MCMC 法によるスパイクソーティングを行った。

シミュレーション 2 の結果のスパイラベルを初期値のスパイクラベルをシミュレーション 2 の結果に設定して、MCMC によるスパイクソーティングの補完でどれだけのパフォーマンスが向上するかを調べた。

### 3.4.1 スパイクシミュレータの設定

スパイクシミュレータの設定は、3.2 節と同じものとした。100 個の全てのスパイクのデータを使った。

### 3.4.2 MCMC の初期設定

ニューロン 1,2,3 パラメータ  $\delta, \lambda$  の事前分布を一様分布で設定した。 $\delta_1, \delta_2, \delta_3$  の事前分布の定義域を 0.1 から 0.9 とした。 $\lambda_1, \lambda_2, \lambda_3$  の事前分布を定義域を 1 から 200 とした。

$$\delta_1, \delta_2, \delta_3 \subset u(0.1, 0.8) \quad (3.3)$$

$$\lambda_1, \lambda_2, \lambda_3 \subset u(1, 200) \quad (3.4)$$

パラメータの初期値を  $\delta = 0.5, \lambda = 100.5$  とした。スパイラベルの初期値はシミュレーション 2 の結果に設定した。パラメータの候補分布には、酔歩過程式 (6.20) をもちいて、 $\tau$  の値をそれぞれの定義域の 0.1 倍とした。スパイクラベルの候補には、現在と異なるラベルを等確率で候補として生成した。各スパイクラベル、パラメータ  $\delta_1, \delta_2, \delta_3, \lambda_1, \lambda_2, \lambda_3$  については各スパイクラベルを調べるごとに 1 回、合計 10000 回、メトロポリス-ヘイスティングスアルゴリズムのステップを反復した。

## 3.5 シミュレーション 5

シミュレーション 4 では、全てのスパイクについてラベルの変更を行った。シミュレーション 5 では、基本的な設定はシミュレーション 4 と同じく、スパイクラベルの初期値は、シミュレーション 2

の結果を用いた。しかし、シミュレーション1と2の結果から、減衰による変動が小さいスパイク (ISI が長いニューロン3からのスパイク) については、ウェーブレット解析によるスパイクソーティングのみで、ソーティング可能と考た。よってシミュレーション5では、シミュレーション2の結果でニューロン3からのスパイクとされたスパイクラベルは、変更しないとした。また、他のスパイクラベルについては、ニューロン3のスパイクラベルに変更しないものとした。

### 3.5.1 スパイクシミュレータの設定

スパイクシミュレータの設定は、シミュレーション4と同じく、3.2節と同じものとした。100個の全てのスパイクのデータを使った。

### 3.5.2 MCMC の設定

初期値の設定は、シミュレーション4と同じである。スパイクラベルの候補には、現在と異なるラベルを等確率で候補として生成した。各スパイクラベル、パラメータ  $\delta_1, \delta_2, \delta_3, \lambda_1, \lambda_2, \lambda_3$  について100回、MCMC法のステップを反復した。

## 4 結果

### 4.1 結果 シミュレーション1

シミュレーションの結果を表すために、Fig. 10、を作った。

10個の基底関数で作られる10次元の特徴空間でのスパイク列をFig. 10(a)に示す。Fig. 10(a)の点一つ一つがスパイクを表す。各スパイクのスパイクラベルについて、ラベル1のスパイクを黒色、ラベル2のスパイクを赤色、ラベル3のスパイクを緑色の点で表している。隣のFig. 10(b)には、10個の基底関数で作られる10次元の特徴空間でのスパイク列をk-means法でクラスタリングした結果を描いた。

Fig. 10の3つの軸、Coeff1, Coeff2, Coeff3は、式(2.13)で、64個の軸の中から標準偏差が1番大きいもの、2番目、3番目のものの3つである。各スパイクのスパイクラベルについて、k-means法の結果、1のクラスターにクラスタリングされたスパイクを黒色、2とクラスタリングされたスパイクを赤色、3とクラスタリングされたスパイクを緑色の点で表している。

クラスタリングの結果を評価するために、Table 4(a), (b), (c)を作った。Table 4に、各スパイクについて、行に正解のスパイクラベル、列にクラスターの番号を割り当てた。正解のスパイクラベルがどのクラスターの番号に対応しているのかが分かる。仮に、全てのスパイクラベルについて、正しくソーティングできたとすると、Table 4(a)のように、対角線上にスパイクがカウントされる。Table 4(a)と(b)を見比べると、スパイクラベル1がクラスター3に、スパイクラベル2がクラスター1に、スパイクラベル3がクラスター2に対応していると解釈できる。

このシミュレーションは、Table 1計算機[A]で1.36(s)かかった。

## 4.2 結果 シミュレーション 2

前節と同じく、10 個の基底関数で作られる 10 次元の特徴空間でのスパイク列を Fig. 13(a)、(b) に示す。スパイクラベルとスパイクを表す点の色との対応は、前節と同じである。

隣の Fig. 13 には、10 個の基底関数で作られる 10 次元の特徴空間でのスパイク列を k-means 法でクラスタリングした結果を描いた。

前節と同じく、クラスタリングの結果を評価するために、Table 5、を作った。正解のスパイクラベルがどのクラスターの番号に対応しているのかが分かる。Table 5 と見比べると、スパイクラベル 1 がクラスター 1 に、スパイクラベル 2 がクラスター 2 に、スパイクラベル 3 がクラスター 3 に対応していると解釈できる。

## 4.3 結果 シミュレーション 3

Table 6 に結果を記す。Table 6(a) に 20 個のスパイク  $\mathbf{S}_1 \sim \mathbf{S}_2$  について、初期状態とソーティング後の違いを示した。Table 6(b) に正解のスパイクラベルを Table 6(c) にソーティング後の結果を示した。示した。式 (2.21) の値をスパイクラベルを一つ調べるごとに記録した。Fig. 15 には、対数尤度式 (2.21) の全 10000 ステップの遷移を描いた。また Fig. 16 には、全ステップの後半 1 割 (9001~10000) のパラメータ  $\delta_{i=1,2,3}, \lambda_{i=1,2,3}$  の標本から、それぞれの事後分布のヒストグラムを作った。

このシミュレーションは、Table 1 計算機 [B] で 10753.19(s) かかった。

## 4.4 結果 シミュレーション 4

Table 7 に結果を記す。Table 7(a) と Table 7(c) を見比べると、スパイクラベル 1 がクラスター 1 に、スパイクラベル 2 がクラスター 3 に、スパイクラベル 3 がクラスター 3 に対応していると解釈できる。Table 7 と Table 7 を見比べると、シミュレーション 2 で正答率が高かったスパイクラベル 3 のスパイクについて、多く間違っている事が分かる。

対数尤度 (式 (2.21)) の値をスパイクラベルを一つ調べるごとに記録した。Fig. 17 に対数尤度の全ステップの遷移を描いた。また Fig. 20 には、全ステップの後半 1 割 (29001~30000) のパラメータ  $\delta_{i=1,2,3}, \lambda_{i=1,2,3}$  の標本から、それぞれの事後分布のヒストグラムを作った。このシミュレーションは、Table 1 計算機 [B] で 272017.6(s) かかった。

## 4.5 結果 シミュレーション 5

Table 8 に結果を記す。Table 8 を見比べると、スパイクラベル 1 がクラスター 1 に、スパイクラベル 2 がクラスター 3 に、スパイクラベル 3 がクラスター 3 に対応していると解釈できる。Table 8 にスパイクラベルを変更するか、しないのかを選択する試行を、91~100 回繰り返した後の結果を載せた。

対数尤度式 (2.21) の値をスパイクラベルを一つ調べるごとに記録した。Fig. 17 に対数尤度 (式 (??)) の全 30000 ステップの遷移を描いた。また Fig. 20 には、全ステップの後半 1 割 (29001~30000) のパラメータ  $\delta_{i=1,2,3}, \lambda_{i=1,2,3}$  の標本から、それぞれの事後分布ヒストグラムを作った。

このシミュレーションは、Table 1 計算機 [B] で 195786.3(s) かかった。



## 5 考察

### 5.1 ウェーブレット解析によるスパイクソーティングについて

シミュレーション 1,2 では、ウェーブレット変換を用いたスパイクソーティングについて、減衰によるスパイク電位の変動が小さい場合（シミュレーション 1）と、減衰によるスパイク電位の変動が大きい場合（シミュレーション 2）との 2 つの場合での、スパイクソーティングのパフォーマンスを調べた。シミュレーション 1 では、ほぼ完璧にソーティングできたが、波形そのものが変わるシミュレーション 2 では、減衰の変動が大きいスパイクについてソーティングの精度は高くなかった。

### 5.2 マルコフ連鎖モンテカルロ法によるスパイクソーティングについて

シミュレーション 3,4,5 では、減衰によるスパイク電位の変動が大きい場合について、MCMC を用いたソーティングをおこなった。

シミュレーション 3 では、20 個のスパイク列について、初期設定のスパイクラベルをランダムな状態から始めたときについて、スパイクソーティングのパフォーマンスを調べた。結果は、スパイク電位の減衰のみを手がかりにしても、ある程度は、スパイクソーティングの精度はあるといえる。この手法はウェーブレット変換によるスパイクソーティングとは、独立した手法であるから、減衰の変動が大きいスパイクをソーティングするための有効な手法であると思われる。

シミュレーション 4 では、100 個のスパイク列について、ウェーブレット変換によるスパイクソーティングを補完するために、シミュレーション 2 の結果のスパイクラベルを初期設定とし各スパイクラベルについての 100 回ずつのサンプリングを行った。結果は、ウェーブレット変換を用いたスパイクソーティング手法では、判別できていた減衰によるスパイク電位の変動が小さいスパイクについて、他の変動が大きいスパイクと間違えてソーティングケースが多かった。

シミュレーション 5 では、単に初期設定にシミュレーション 2 の結果のスパイクラベルを設定するだけでなく、シミュレーション 2 で決定したスパイクラベルについて、ISI が長いニューロン（減衰によるスパイク電位の変動が小さい）のスパイクについては、スパイクラベルを固定して変更しないとした。結果、91 回目 100 回目のスパイクラベル選択時のソーティング後の結果を記したが、この中には、正解にかなり近い結果（98 回目）もあったが、初期値の状態より悪い結果もあった。これは、10000 回程度のステップの反復では、マルコフ連鎖が定常分布に達していないのか、そもそもスパイク電位の減衰のみをモデルに組み入れたスパイクソーティングの限界ではないか、と様々な原因が考えられる。

また、このスパイクソーティングの手法は非常に時間がかかった。そのため、ステップ数に制限があったので、計算は事後分布が収束するまで続ける事ができなかったことも考えられる。

今後は、レプリカ交換モンテカルロ法を実装する等の工夫をして、計算時間の短縮をする必要がある。

## 6 付録

### 6.1 ウェーブレット解析について

この節を書くために、文献<sup>4) 15)</sup>を参考にした。

#### 6.1.1 フーリエ変換とウェーブレット変換

今回は、スパイク波形から特徴を抽出するためにウェーブレット変換という手法を用いた。ウェーブレット変換とは、信号の時間周波数解析の手法の一つである。ウェーブレット変換を用いると、一次元の信号を周波数分解する事ができる。信号の周波数解析の手法として有名なものには、フーリエ変換がある。信号に対しフーリエ変換を用いる事により、信号を指数関数の和に分解し、その信号にどれだけの周波数の波が含まれるのかを調べる事ができる。フーリエ変換が指数関数の和として信号を分解するのに対し、ウェーブレット変換は、ウェーブレット（小さい波）と呼ばれる特殊な関数の和で信号を分解する。フーリエ変換とウェーブレット変換を数式にして見比べると以下になる。（連続）フーリエ変換

$$f(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} d\omega \quad (6.1)$$

（連続）ウェーブレット変換

$$W(\omega) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt \quad (6.2)$$

$$\psi_{a,b}(t) = \psi\left(\frac{t-b}{a}\right) \quad (6.3)$$

フーリエ変換の基底関数は周期が連続した指数関数である。これに対し、ウェーブレット変換の基底関数  $\psi(x)$  は、空間に局所的な関数である。ウェーブレット関数はフーリエ変換の基底関数とは異なり、様々な種類のものがある。今回は、Daubechies の基底関数と呼ばれるウェーブレット関数をスパイクの特徴抽出に採用した。Daubechies の基底関数 (Fig. 4) は以下の性質を持つ、ウェーブレット関数である。(1) コンパクトサポートを持つ (平たく言うと、一つの区間でのみ値をもち、他での値は 0)、(2) 正規直交系。性質 (2) によって、ウェーブレット関数を用くと、多重解像度解析 (Fig. 3) と呼ばれる離散ウェーブレット変換のアルゴリズムを使用できる。

#### 6.1.2 多重解像度解析

以下は、離散ウェーブレット変換のアルゴリズムの一つである多重解像度解析 (Fig. 3) について説明する。多重解像度解析とは、信号处理的に言うと、解析する信号にローパスフィルタとバンドパスフィルタ（ハイパスフィルタ）をかけ、ローパスフィルタで取りだした低域成分にさらにローパスフィルタとバンドパスフィルタをかける、ということを繰り返すことによって、離散ウェーブレット変換を行うアルゴリズムである。元の信号を level 0 とし、フィルタをかけるごとに、信号を level -1, level -2, ..., と呼ぶ。データの要素数が  $2^x$  である信号は最大で level -x まで多重解像度解析を行う事ができる。

多重解像度解析では、ウェーブレット変換の基底関数であるウェーブレット関数  $\psi(x)$  を、次のようにスケーリング関数  $\phi(x)$  から構成する。

スケーリング関数

$$\phi_{j,n}(t) = \sqrt{2}^j \phi(2^j t - n) \quad (6.4)$$

ウェーブレット関数

$$\psi_{j,n}(t) = \sqrt{2}^j \psi(2^j t - n) \quad (6.5)$$

多重解像度解析を信号処理的に言うと、スケーリング関数がローパスフィルタ、ウェーブレット関数がバンドパスフィルタ（ハイパスフィルタ）である。

ここで信号  $f$  は次のように  $\phi(t)$  で補完する事ができる。

$$f(t) = \sum_m c_{0,m} \phi(t - m) \quad (6.6)$$

level 0 の信号の多重解像度解析の中身はこの  $c_{0,m}$  の列、 $\{c_{0,0}, c_{0,1}, \dots, c_{0,2^x}\}$  スケーリング関数とウェーブレット関数は次のツースケール数列と呼ばれる  $p_n, q_n$  に関して、ツースケール関係と呼ばれる関係を満たす。

ツースケール数列  $p_n, q_n$  は次のような数列である。

$$q_n = (-1)^{1-n} p_{1-n} \quad (6.7)$$

$$\sum_n p_n = 2 \quad (6.8)$$

このツースケール数列具体的な値は、ウェーブレット関数の作者によって公開されている。ドベシーはカスケードアルゴリズムという方法で、ツースケール数列を求めた。ツースケール数列は  $\phi(t), \psi(t)$  と、次のツースケール関係をみだす。

$$\phi(t) = \sum_n p_n \phi(2t - n) \quad (6.9)$$

$$\psi(t) = \sum_n q_n \phi(2t - n) \quad (6.10)$$

このツースケール数列から、次の分解数列  $h_0[n], h_1[n]$  が

$$h_0[n] = \frac{1}{\sqrt{2}} \bar{p}_{-n} \quad (6.11)$$

$$h_1[n] = \frac{1}{\sqrt{2}} \bar{p}_{-n} \quad (6.12)$$

と定義され、分解数列  $h_0[n], h_1[n]$  を用いる事によって次の分解アルゴリズム

$$c_{j-1,n} = \sum_m h_0[2n - m] c_{j,m} \quad (6.13)$$

$$d_{j-1,n} = \sum_m h_1[2n - m] c_{j,m} \quad (6.14)$$

が計算できる。分解アルゴリズムから計算した、ウェーブレット係数  $d_{-1,n}, d_{-2,n}, \dots, d_{-j,n}$  とスケーリング関数係数  $c_{-j,n}$  が Level-j で多重解像度解析した後の信号のデータの中身である。

## 6.2 k-means 法

分類するデータを  $n$  個とし、 $\mathbf{x}_i (i = 0 \sim n)$  とする。各データは  $j$  次元とし、 $\mathbf{x}_i = (x_0, x_1, \dots, x_{j-1})$  とする。クラスターの数を  $k$  個とする。データ間の距離を定義する。データの全ての点を、有限個の代表点と距離に基づいて、最も近い点を集めて作った領域を Voronoi 領域という。

k-means 法の手順は、次のようになる。

- (1) 各データの中から、 $k$  個の代表ベクトルをランダムに選んで初期値とする。
- (2) 代表ベクトルの Voronoi 領域を求め、各 Voronoi 領域の平均値を計算する。平均値に一番近いデータを新しい代表ベクトルとする。
- (3) 代表ベクトルが収束するまで、(2) を繰り返す。

## 6.3 ベイズ統計とマルコフ連鎖モンテカルロ法

この節を書くために文献<sup>1)17)</sup>を参考にした。

### 6.3.1 ベイズ統計とベイズの定理

ベイズ統計では、不確実な現象の背後にあるパラメータ  $\theta$  の確率分布である事前分布を仮定した上で、確率分布の性質を決定するパラメータの値を観測されたデータに基づいて事後分布を推測する。事後分布は、ベイズの定理により事前分布と尤度関数から計算する事ができる。 $\mathbf{x}$  がパラメータ  $\theta$  もつ、ある確率分布  $X(\theta)$  から生成された観測データであるとする。 $\mathbf{x} \sim X(\theta)$  となる観測値  $\mathbf{x}$  未知のパラメータ  $\theta$  に関する情報を事前情報といい、その確率分布を事前分布と呼ぶ。その確率密度関数と呼び、 $\pi(\theta)$  で表す。尤度関数  $f(\mathbf{x}|\theta)$  観測値  $\mathbf{x}$  を得た後の  $\mu$  の分布を事後分布と呼び、その確率密度関数  $\pi(\theta|\mathbf{x})$  を事後確率密度関数と呼ぶ。

ベイズの定理によると、事後確率密度関数は、事前確率密度関数と尤度関数から次のように計算できる。

$$\pi(\theta|x) = \frac{\pi(x, \theta)}{\int P(x, \theta) d\theta} \quad (6.15)$$

$$\propto f(\mathbf{x}|\theta)\pi(\theta) \quad (6.16)$$

となる。これにより、曖昧な（分散の大きい）事前分布からも、限定された（分散の小さい）事後分布を計算する事ができる。

この事後分布からのサンプルを生成するために、マルコフ連鎖モンテカルロ法が用いられる。

### 6.3.2 マルコフ連鎖モンテカルロ法 (MCMC)

マルコフ連鎖モンテカルロ法 (MCMC) を用いると、一様分布や正規分布だけからではなく、様々な多変量な確率分布からサンプルを生成する事ができる。ベイズ統計では、マルコフ連鎖モンテカルロ法が用いることによって、事後分布  $P(\theta|x)$  からの乱数を生成する事ができる。

あるシステムがあつて、そのシステムは様々な状態をとるとする。マルコフ連鎖とは、一般にあるシステムの状態が  $\mathbf{x}$  となっていて、次の状態が直前の状態  $\mathbf{x}'$  のみに依存して決定するシステムのことである。状態  $\mathbf{x}$  から次の状態  $\mathbf{x}'$  に移る事を  $\mathbf{x} \rightarrow \mathbf{x}'$  と書き、その確率（遷移確率）を  $\pi(\mathbf{x} \rightarrow \mathbf{x}')$  と書

く事にする。マルコフ連鎖は、正則条件と呼ばれる条件が満たされたときに、マルコフ連鎖の反復を繰り返していくと、サンプルが定常分布に収束することが知られている。このとき、 $P(\mathbf{x})$  を遷移確率  $p$  で決まるマルコフ連鎖の定常分布になっていると言う。マルコフ連鎖からサンプルを生成したい確率分布を  $P(\mathbf{x})$  とし、これを目標分布と呼ぶ。

目標分布が定常分布になるように、マルコフ連鎖を構成することにより、マルコフ連鎖を用いて目標分布からの確率標本をえることができる。このようなマルコフ連鎖を次のメトロポリス-ヘイスティングスアルゴリズムで構成することができる。

### 6.3.3 メトロポリス-ヘイスティングスアルゴリズム

メトロポリス-ヘイスティングスアルゴリズムを使うと、計算機から容易に生成する事ができない事後分布  $\pi^*$  からの乱数を、事後分布を近似する提案分布  $q^*$  を使う事で生成することができる。提案分布は計算機からの乱数の生成が簡単なものを選ぶ。

事後分布の確率密度関数を  $\pi(\theta|\mathbf{x})$ , 提案分布の確率密度関数を  $q(\theta|\mathbf{x})$  とする。パラメータ  $\theta = (\theta_1, \theta_2, \dots, \theta_p)$  とし、 $t$  ステップ目のパラメータを  $\theta^{(t)}$  で表す。

- (1) 初期値  $\theta^{(0)}$  を決める。
- (2)  $i = 1, 2, \dots, p$  について、以下を繰り返す。 現在  $\theta_i^{(t-1)}$  であるとき、次の状態  $\theta_i^t$  の候補  $\theta'_i$  を提案分布  $q_i(\theta_i^{(t-1)}, \theta_{-i}|\mathbf{x})$  から発生させる。

$$\alpha_i(\theta_i^{(t-1)}, \theta'_i|\theta_{-i}^{(t)}, \mathbf{x}) = \begin{cases} \min\left\{\frac{\pi(\theta_i|\theta_{-i}^{(t)}, \mathbf{x})q(\theta_i^{(t-1)}|\theta_{-i}^{(t)}, \mathbf{x})}{\pi(\theta_i^{(t-1)}|\theta_{-i}^{(t)}, \mathbf{x})q(\theta_i|\theta_{-i}^{(t)}, \mathbf{x})}, 1\right\}, & (\pi(\theta_i^{(t-1)}|\theta_{-i}^{(t)}, \mathbf{x})q_i(\theta_i^{(t-1)}|\theta_{-i}^{(t)}, \mathbf{x}) \geq 0) \\ 1, & (\pi(\theta_i^{(t-1)}|\theta_{-i}^{(t)}, \mathbf{x})q_i(\theta_i^{(t-1)}|\theta_{-i}^{(t)}, \mathbf{x}) = 0) \end{cases} \quad (6.17)$$

ここで、 $\theta_{-i}^t = (\theta_1^t, \dots, \theta_i^t, \theta_{i+1}^{(t-1)}, \dots, \theta_p^{(t)})$  である。

- (3) (0,1) 上の一様乱数  $u$  を発生させて、

$$\theta_i^{(t)} = \begin{cases} \theta'_i, & u \leq \alpha_i(\theta_i^{(t-1)}, \theta'_i|\theta_{-i}^{(t)}, \mathbf{x}) \text{ のとき} \\ \theta_i^{(t-1)}, & u \geq \alpha_i(\theta_i^{(t-1)}, \theta'_i|\theta_{-i}^{(t)}, \mathbf{x}) \text{ のとき} \end{cases} \quad (6.18)$$

とする。

- (4)  $t$  を  $t+1$  として (2) にもどる。

#### 6.3.3.1 酔歩過程

事後分布を近似するような提案分布を探す事は難しい。そこで、パラメータ  $\mu$  の新しい候補に、次の酔歩過程を提案分布として用いる事ができる。

$$\mu' = \mu^{t-1} + \epsilon \quad (6.19)$$

$$\epsilon \sim N(0, \tau^2) \quad (6.20)$$

## 謝辞

本研究は大阪大学大学院生命機能研究科脳神経工学講座大澤研究室で行われた。本研究を遂行するにあたり、多大なる御指導そして御協力を賜りました主査の大澤五住教授に心より感謝します。副査の菊池誠教授、並びに貴重な助言をしてくださった小林康助教授、田中宏喜助手、畑中信技官に深く感謝します。最後に様々な助言、協力を頂いた真田さんをはじめとする大澤研究室の皆様に、この場を借りて感謝の意を表します。

## 参考文献

- 1) 伊庭幸人種村正美, 大森裕浩, 和合肇, 佐藤整尚, 高橋明彦. 計算統計2 マルコフ連鎖モンテカルロ法とその周辺. 岩波書店, 2005.
- 2) Pascal Viot Christophe Pouzat, Matthieu Delescluse and Jean Diebolt. Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: A markov chain monte carlo approach. *J Neurophysiol*, Vol. 91, pp. 2910–2928, 2004.
- 3) Ben-Jacob E Hulata E, Segev R. A method for spike sorting and detection based on wavelet packets and shannon mutual information. *J. Neurosci.Methods*, Vol. 117, pp. 1–12, 2004.
- 4) I. ドブシー. ウェーブレット10講. シュプリンガー・フェアラーク東京, 2003.
- 5) P. P. Mitra M. S. Fee and D. Kleinfeld. Variability of extracellular spike waveforms of cortical neurons. *J Neurophysiol*, Vol. 76, pp. 3823–3833,, 1996.
- 6) Matsumoto and Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. on Modeling and Computer Simulation*, Vol. 8, pp. 3–30, 1998.
- 7) Lewicki MS. A review of methods for spike-sorting:the detection and classification of neural action potentials. *Network:Comput Neural Syst*, Vol. 9, pp. 53–78, 1998.
- 8) Brown EN Nguyen DP, Frank LM. An application of reversible-jump markov chain mote carlo to spike classification of multi-unit extracellular recordings. *Network:Comput neural Syst*, Vol. 14, pp. 61–82, 2003.
- 9) Christophe Pouzat. Technique(s) for spike-sorting. *q-bio.QM/0405012*, 2004.
- 10) Laurent G Pouzat C, Mazor O. Using noise signature to optimize spike - sorting and to asses neuronal classification quality. *J Neurosci Methods*, Vol. 122, pp. 43–57, 2004.
- 11) Laurent G Pouzat C, Mazor O. Using noise signature to optimize spike - sorting and to asses neuronal classification quality. *J Neurosci Methods*, Vol. 122, pp. 43–57, 2004.
- 12) Nadasdy Z Quian Quiroga R and Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput*, Vol. 16, pp. 1661–87, 2004.
- 13) R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.
- 14) 吉川武樹. バースト発火中のスパイクについて. 研究室修士論文, Vol. kari, p. kari, 2004.
- 15) 戸田浩, 章忠, 川畑洋昭. 最新ウェーブレット実践講座. ソフトバンククリエイティブ, 2005.
- 16) 森屋彰久. kari. kari, Vol. kari, p. kari, 2003.
- 17) 中妻照雄. ファイナンスのためのMCMC法によるベイズ分析. (財) 三菱経済研究所, 2003.

## 付 図

1	実験システム . . . . .	1
2	ウェーブレット解析によるスパイクソーティング . . . . .	2
3	多重解像度解析 . . . . .	3
4	Daunbecies の基底関数 . . . . .	4
5	減衰する電位 . . . . .	4
6	MCMC によるスパイクソーティング (1) . . . . .	4
7	スパイク電位の減衰の確率モデル . . . . .	5
8	MCMC によるスパイクソーティング (2) . . . . .	5
9	simulation 1: それぞれのテンプレート波形 . . . . .	6
10	simulation 1: ウェーブレット変換されたスパイク列 . . . . .	7
11	simulation 1: 基底関数 $\phi_1 \sim \phi_{64}$ の得点 . . . . .	7
12	simulation 2: それぞれのテンプレート波形 . . . . .	8
13	simulation 2: ウェーブレット変換されたスパイク列 . . . . .	9
14	simulation 2: 基底関数 $\phi_1 \sim \phi_{64}$ の得点 . . . . .	9
15	simulation 3: 対数尤度の遷移 . . . . .	10
16	simulation 3: 各パラメータのヒストグラム . . . . .	10
17	simulation 4: 対数尤度関数の遷移 . . . . .	11
18	simulation 4: 各パラメータのヒストグラム . . . . .	11
19	simulation 5: 対数尤度関数の遷移 . . . . .	12
20	simulation 5: 各パラメータのヒストグラム . . . . .	12

## 付 表

1	使用した計算機 . . . . .	13
2	simulation 1: スパイクデータ . . . . .	13
3	simulation 2: スパイクデータ . . . . .	13
4	simulation 1: 結果 . . . . .	14
5	simulation 2: 結果 . . . . .	14
6	simulation 3: 結果 . . . . .	14
7	simulation 4: 結果 . . . . .	14
8	simulation 5: 結果 . . . . .	15



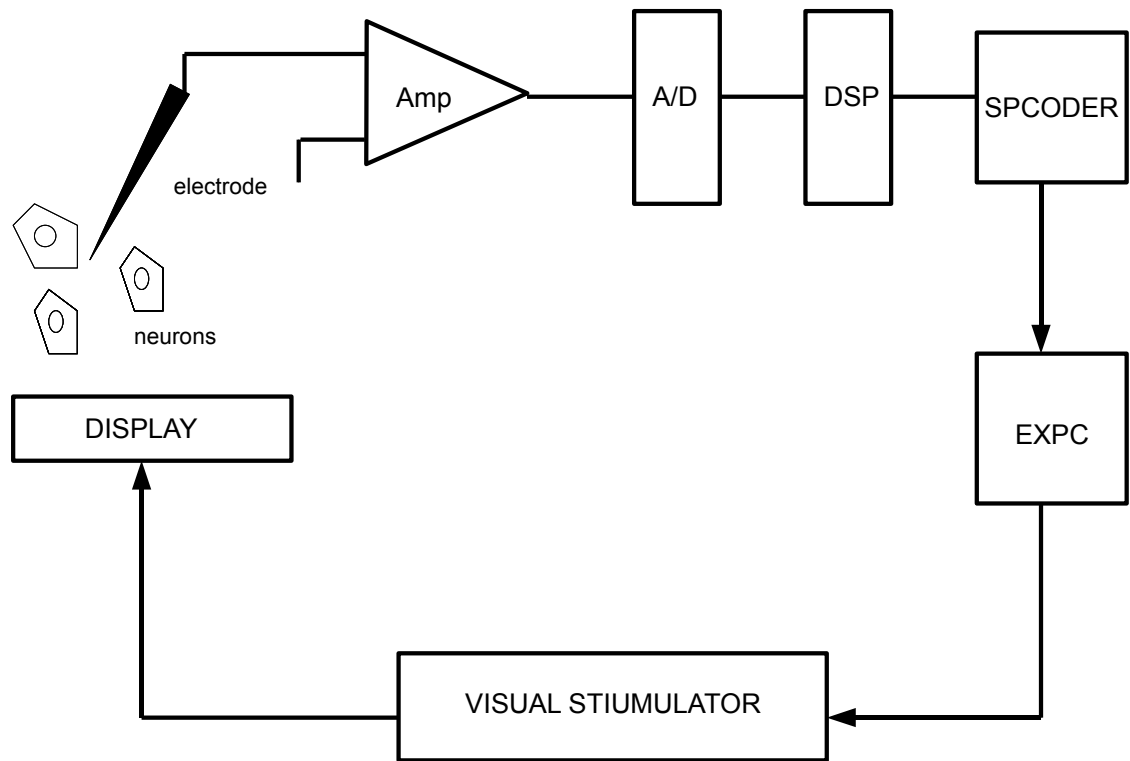
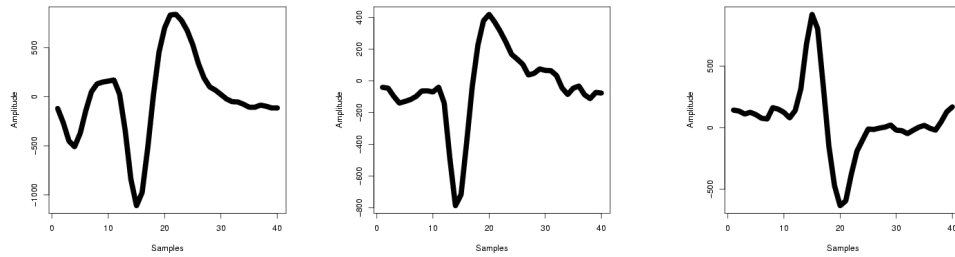


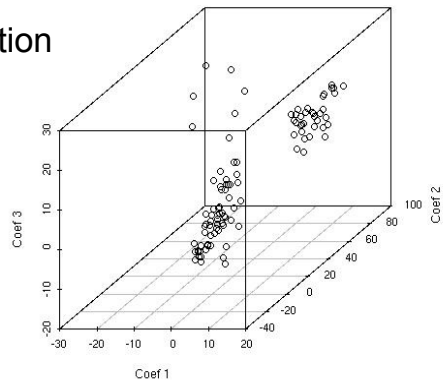
Fig. 1 実験システム

実験システムは、増幅器と3台の計算機 SPCODER（スパイクソーティング用）、EXPC（実験制御用）、VISUAL STIMULATOR（刺激提示用）からなる。被験体に VISUAL STIMULATOR から制御された視覚刺激をモニター (DISPLAY) から提示し、神経細胞の活動電位を単一電極で記録する。活動電位はアンプで約 1000 万倍に増幅し、300 5000Hz のバンドパスフィルタをかけられる。その後 AD 変換器で 25kHz でサンプリングし、12bit に量子化された離散データに変換される。この時、SPCODER で離散データからスパイクが検出される。本研究室の実験システムでは、実験中に実験者がスパイク波形の特徴を自分の目で判別して、手作業にてスパイクソーティングを行っている。スパイクソーティングの結果と各スパイクの波形データ、各スパイクの発火時間は EXPC に送信され、バイナリファイルとして記録される。

## Spikes



## Feature extraction



## k-means clustering

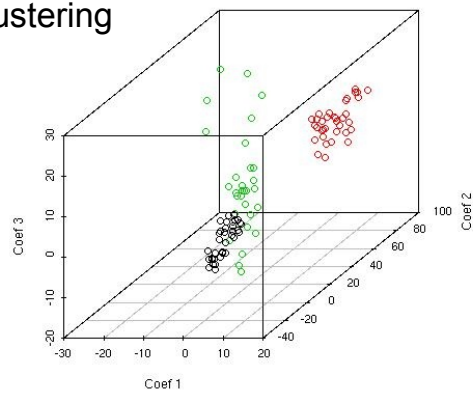


Fig. 2 ウェーブレット解析によるスパイクソーティング

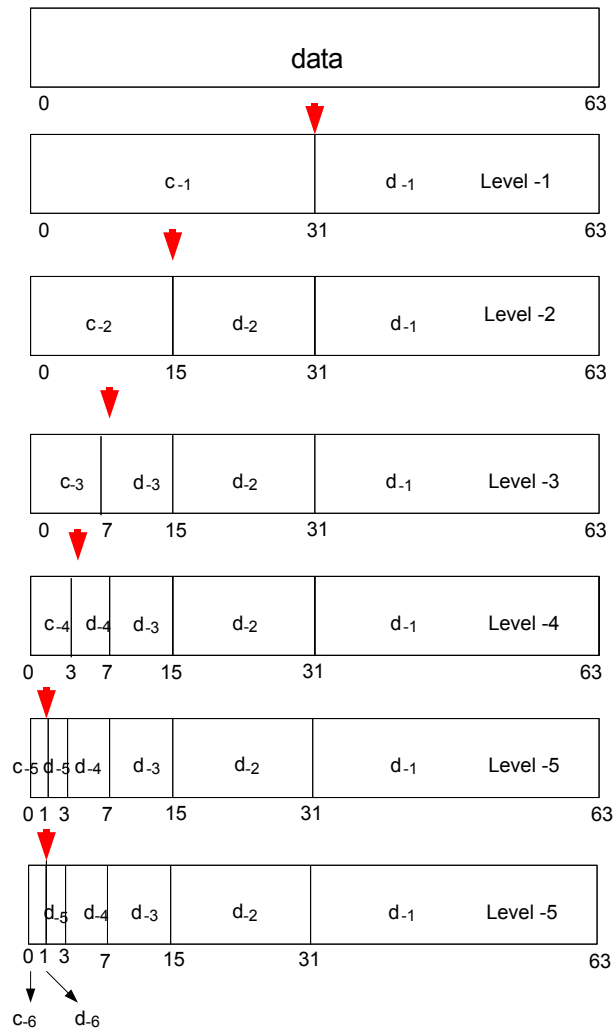


Fig. 3 多重解像度解析

(最上段) 離散ウェーブレット変換をする前の信号のデータ。信号は要素番号を0 63番とした、要素数64個のデータからなるものとする。(上から2段目) 信号をlevel -1の多重解像度解析したデータ。(上から三段目) 信号をlevel -1の多重解像度解析したデータ。多重解像度解析は、周波数の高い方から順に、level-1,level-2,level-3,... 多重解像度解析を行うためには、信号のデータの要素数は2の累乗でなくてはならず、要素数 $2^x$ の信号データは、level-xまで多重解像度解析を行う事ができる。

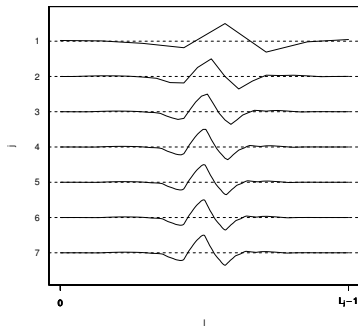


Fig. 4 Daubecies の基底関数

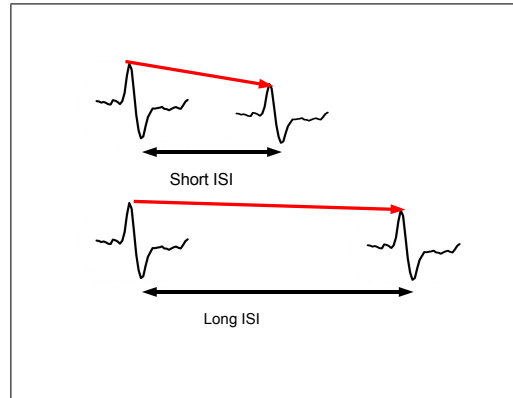


Fig. 5 減衰する電位

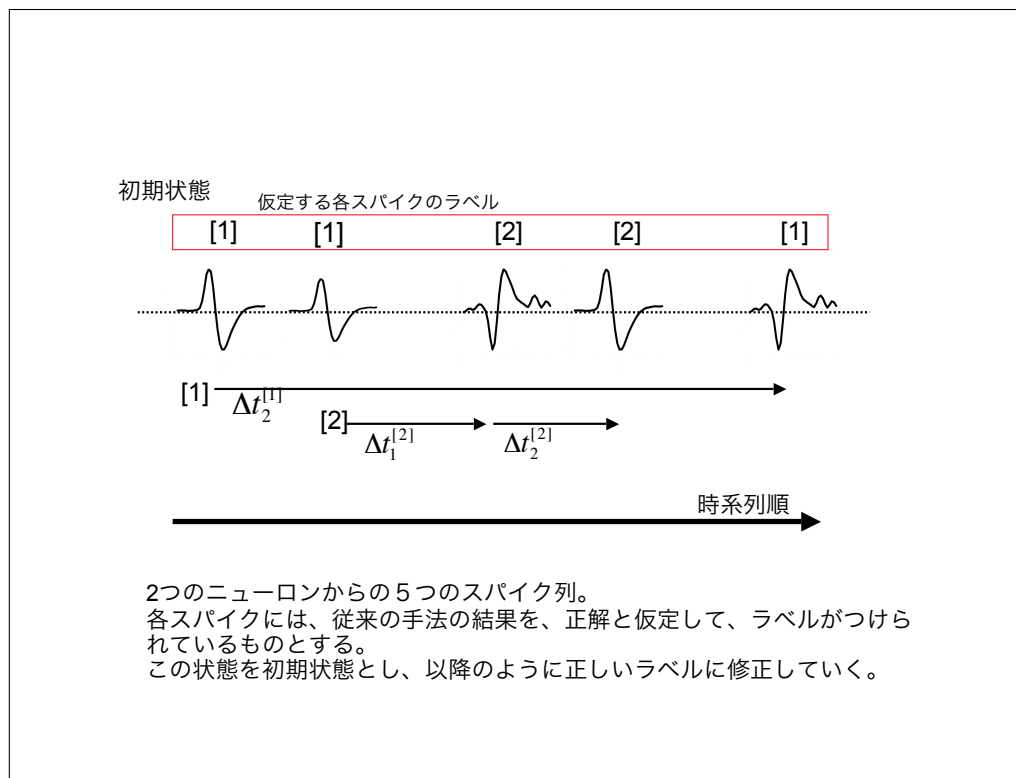


Fig. 6 MCMC によるスパイクソーティング (1)

Fig. 5 上段は、短い ISI におけるスパイク電位の減衰。下段は、長い ISI におけるスパイク電位の減衰。スパイク電位は、ISI が短ければ短いほどスパイク電位の減衰が大きくなる。

Fig. 6 仮に 2 つのニューロン、ニューロン 1 とニューロン 2 から発火された 5 つのスパイクからのなるスパイク列があったとする。MCMC によるスパイクソーティングは、正しいかどうかは別として、適当なスパイクラベルを初期値として仮定し、このスパイクラベルを修正していくステップを反復しながら、進められる。

確率モデルによるスパイクソーティング (2つのスパイクの場合)

- (1) 2つのスパイクについて、実際の電位の値と減衰の確率モデルを比べて、現在のラベルの状態が妥当か、どうかという事を調べる。
- (2) 減衰率が確率的に妥当ならば、ラベルを変更せず、妥当でなければ、ラベルを変更する。

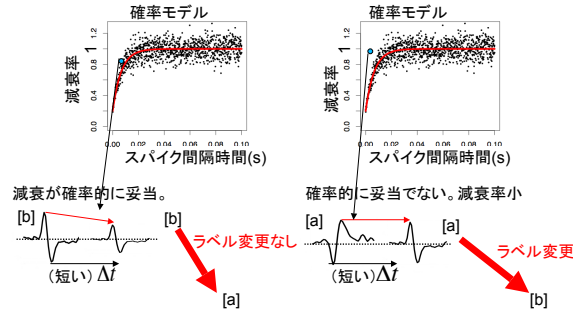


Fig. 7 スパイク電位の減衰の確率モデル

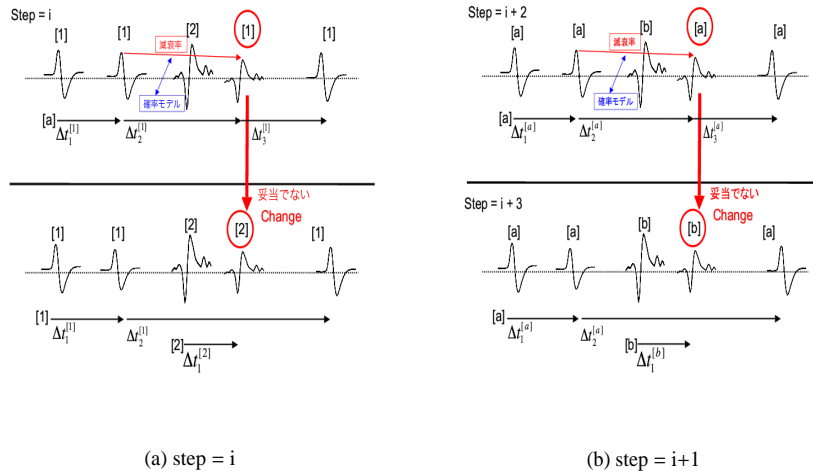


Fig. 8 MCMC によるスパイクソーティング (2)

Fig. 8(a) 現在、step=i として、4 番目のスパイクのラベルを変更するか、しないのかを選択しているとする。今調べているスパイクと同じスパイクラベルを持つ、一つ手前のスパイクから今調べているスパイクの減衰率を調べる。減衰率が確率的に妥当でない場合は、ラベルを変更する。step=i+1 として、次のスパイクを調べる。

Fig. 8(b) 現在、step=i+1 として、5 番目のスパイクのラベルを変更するか、しないのかを選択しているとする。今調べているスパイクと同じスパイクラベルを持つ、一つ手前のスパイクから今調べているスパイクの減衰率を調べる。減衰率が確率的に妥当な場合は、ラベルを変更しない。このように順々に全てのスパイクを調べていく。

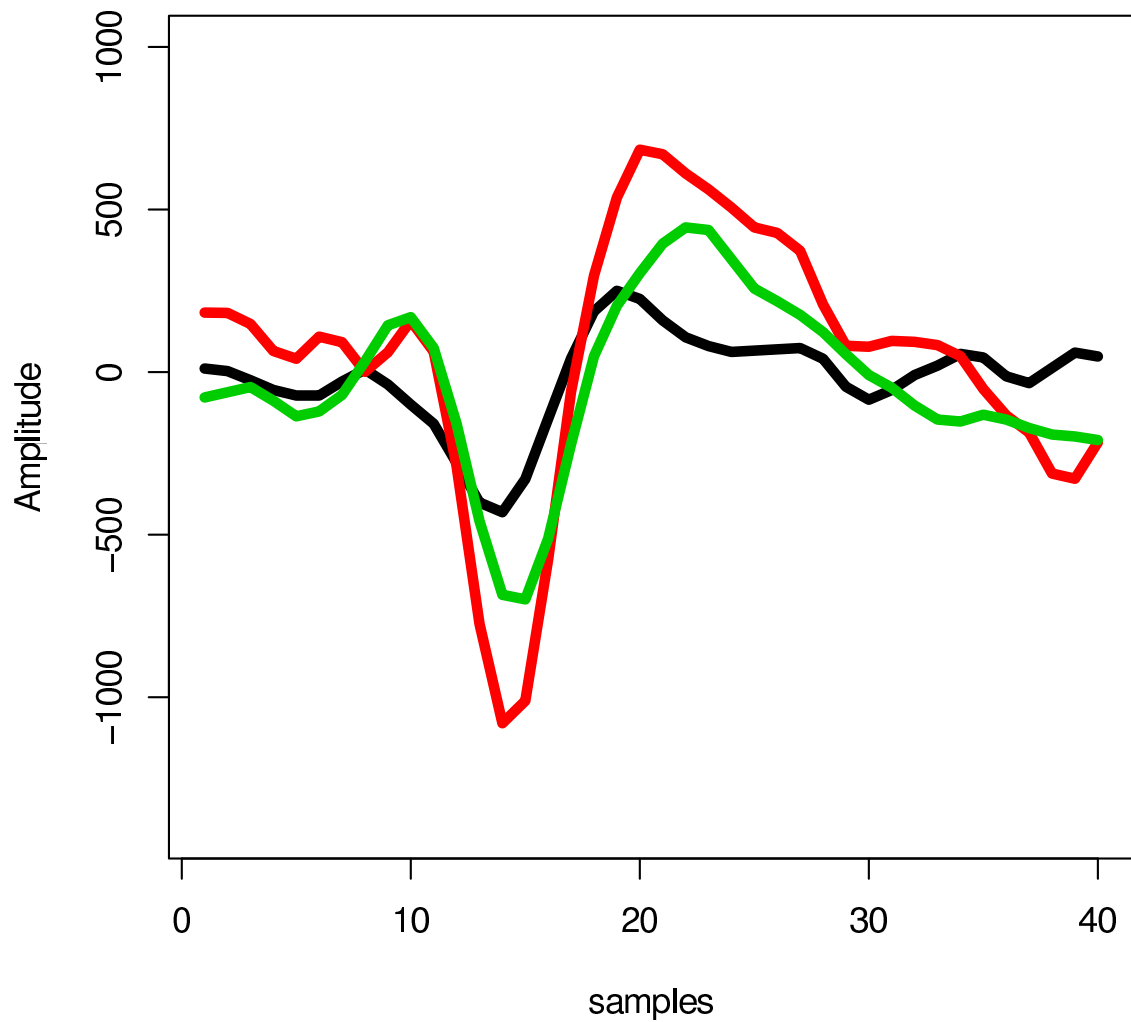
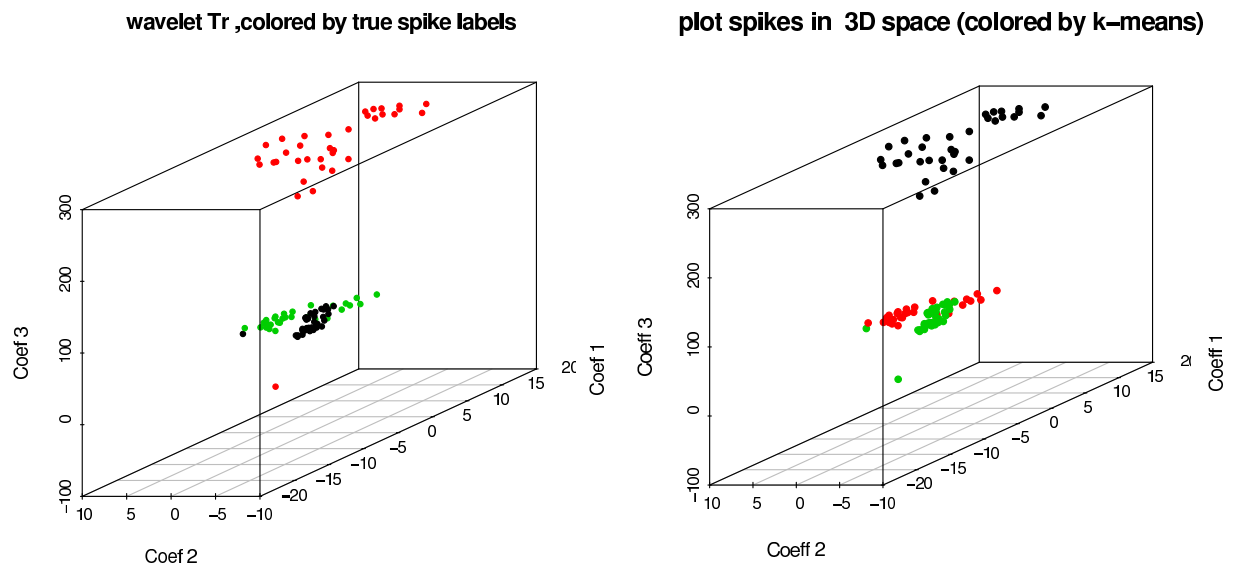


Fig. 9 simulation 1: それぞれのテンプレート波形

各スパイク波形は、次の EV ファイルの中から、波形を読み込んだものである。

- 黒:04-9A\_ep013.005.ev
- 赤:03-35Bep008.004.ev
- 緑:0310d\_ep031.003.ev



(a) 正しいラベル

(b) ソーティング後

Fig. 10 simulation 1: ウェーブレット変換されたスパイク列

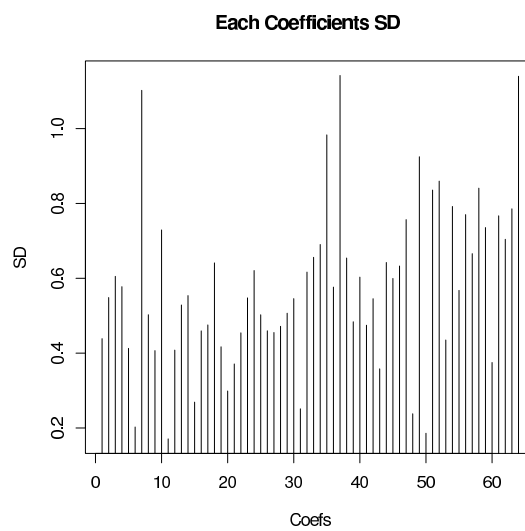


Fig. 11 simulation 1: 基底関数  $\phi_1 \sim \phi_{64}$  の得点

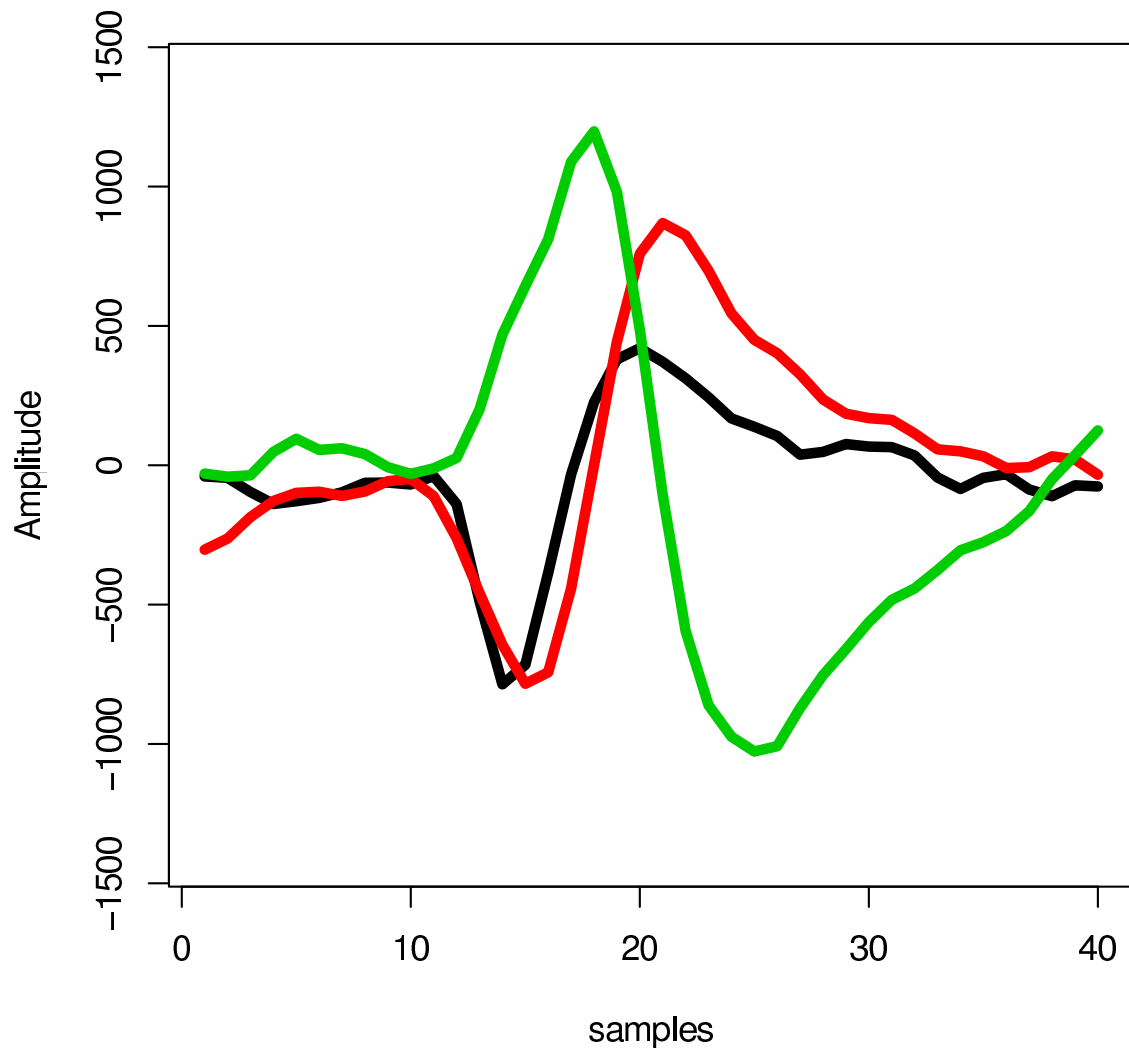


Fig. 12 simulation 2: それぞれのテンプレート波形

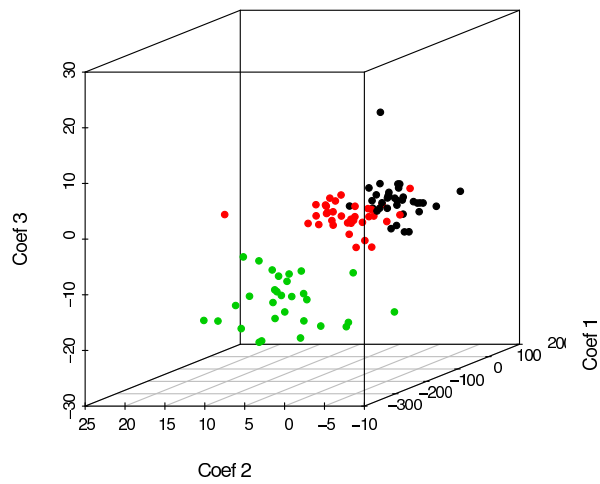
各スパイク波形は、次の EV ファイルの中から、波形を読み込んだものである。

- 黒:rkcy00ep050.001.ev
- 赤:0318aep005.011.ev
- 緑:05-9Cep031.011.ev

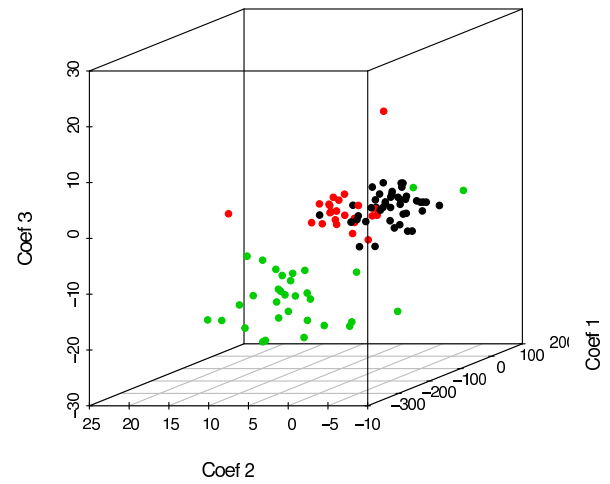


## シミュレーション 2

plot spikes in 3D space (colored by answer)



plot spikes in 3D space (colored by k-means)



(a) 正解

(b) ソーティング後

Fig. 13 simulation 2: ウェーブレット変換されたスパイク列

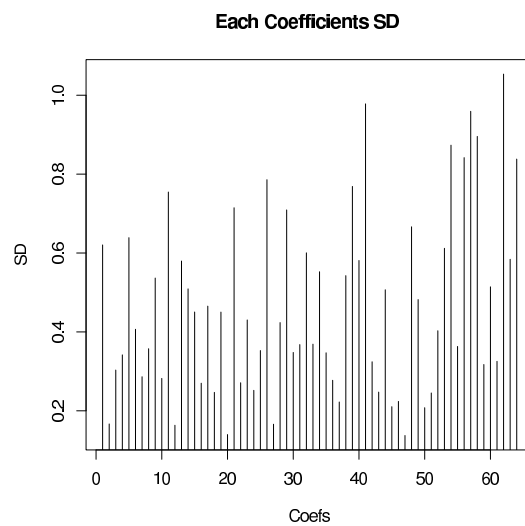


Fig. 14 simulation 2: 基底関数  $\phi_1 \sim \phi_{64}$  の得点

シミュレーション3

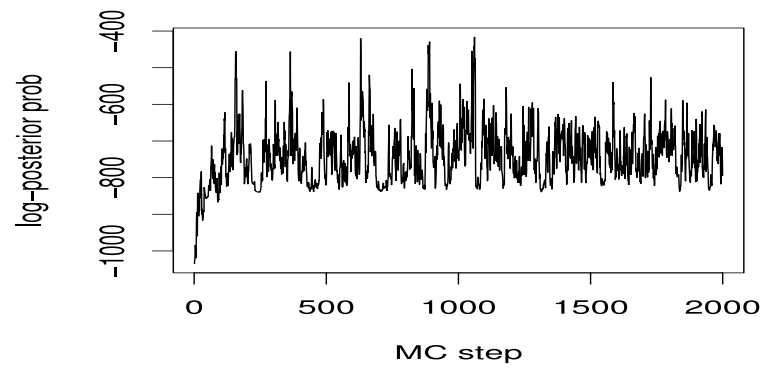


Fig. 15 simulation 3: 対数尤度の遷移

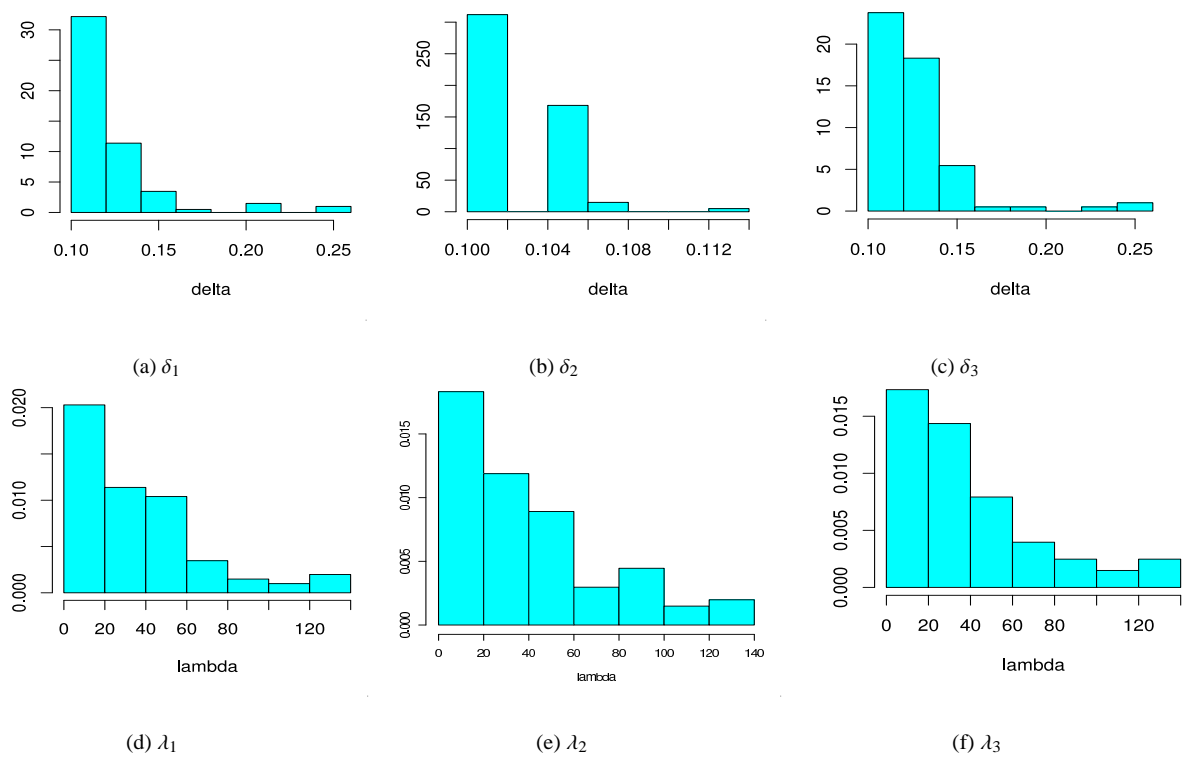


Fig. 16 simulation 3: 各パラメータのヒストグラム

シミュレーション 4

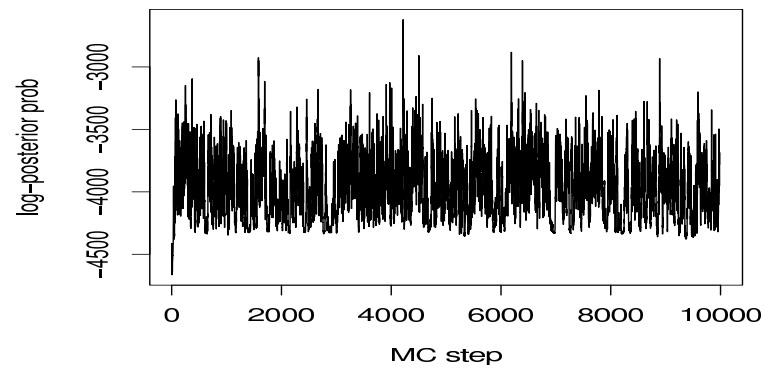


Fig. 17 simulation 4: 対数尤度関数の遷移

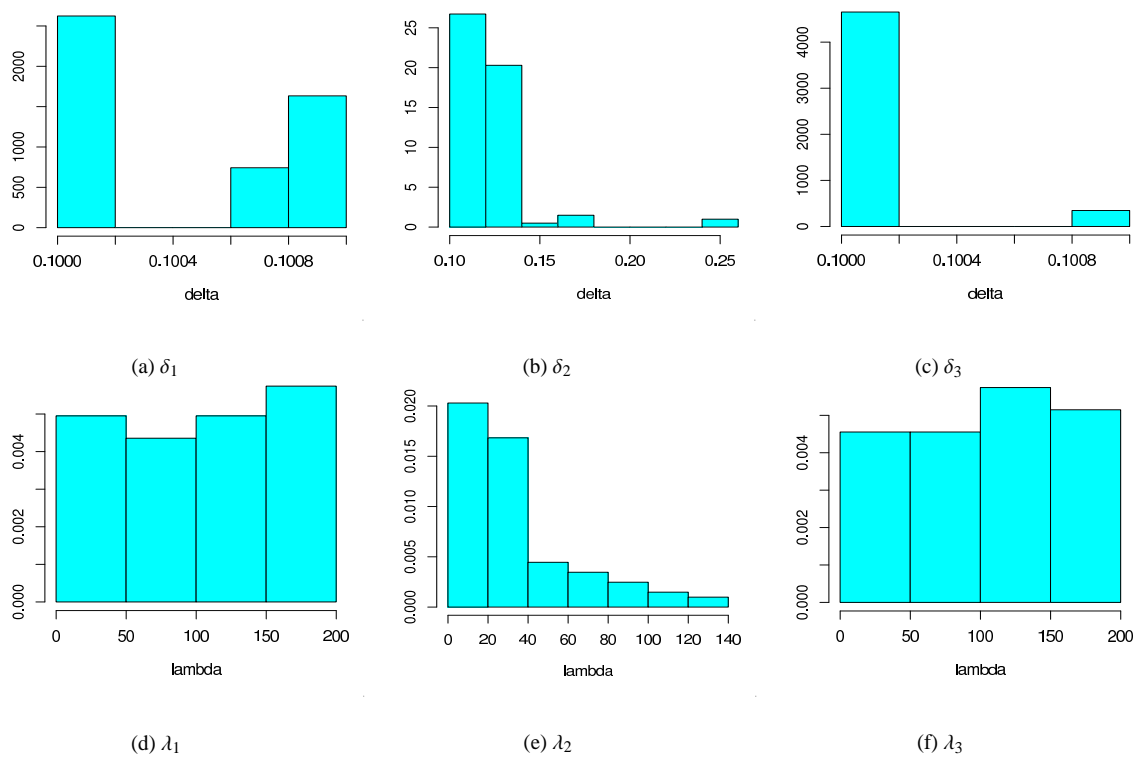


Fig. 18 simulation 4: 各パラメータのヒストグラム

シミュレーション 5

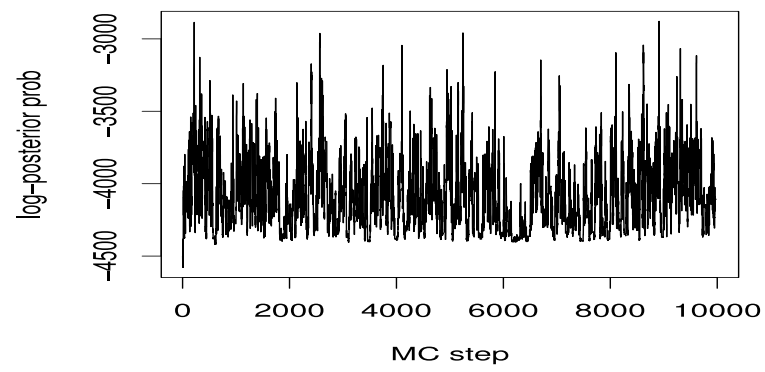


Fig. 19 simulation 5: 対数尤度関数の遷移

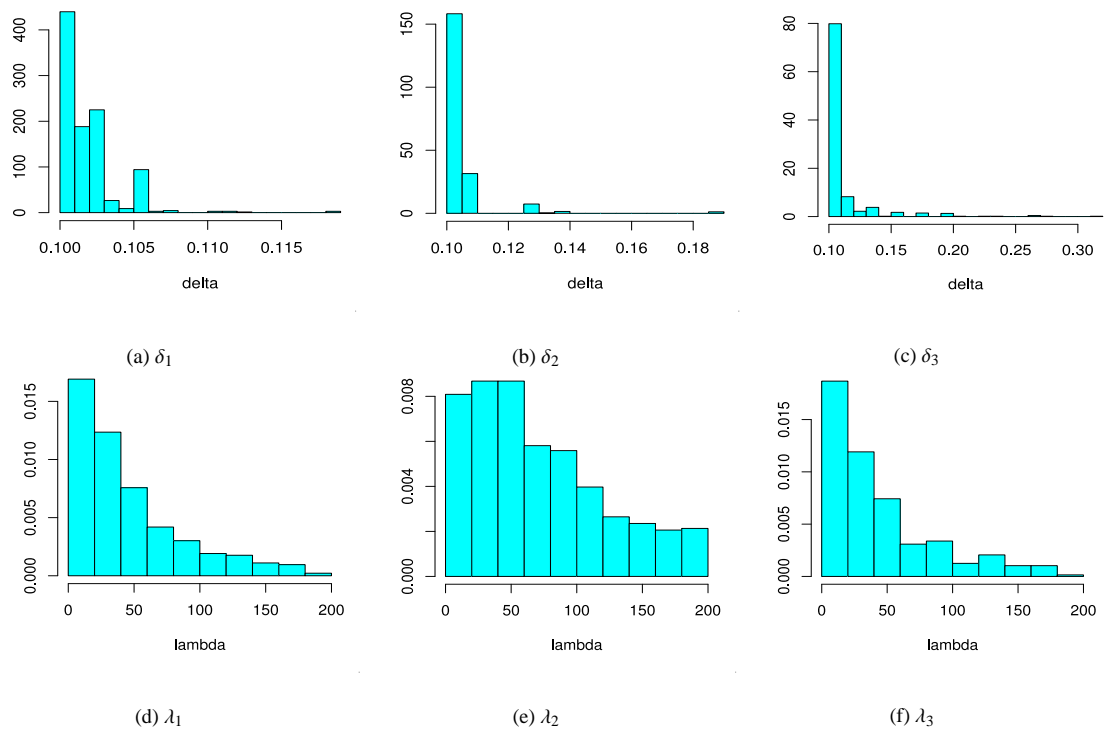


Fig. 20 simulation 5: 各パラメータのヒストグラム

Table 1 使用した計算機

	[A]	[B]
Machine Model	iBook G4	RackMac3.1
CPU Type	PowerPC G4(1.2)	PowerPC G5(3.1)
Number of CPU	1	2
CPU Speed	1.33GHz	2.3GHz
memory	1.25GB	4GB
Bus Speed	133MHz	1.15GHz

Table 2 simulation 1: スパイクデータ

	テンプレート	スパイクの数	$\delta$	$\lambda$	平均発火率 (s)
ニューロン 1	rkcy00ep050.001.ev	35	0.8	100	0.150
ニューロン 2	0318aep005.011.ev	35	0.8	100	0.170
ニューロン 3	05-9Cep031.011.ev	30	0.8	150	0.200

Table 3 simulation 2: スパイクデータ

	テンプレート	スパイクの数	$\delta$	$\lambda$	平均発火率 (s)
ニューロン 1	038c__ep025.006.ev	35	0.5	180	0.020
ニューロン 2	03-20cep036.015.ev	35	0.5	150	0.035
ニューロン 3	043c__ep012.010.ev	30	0.5	150	0.200

Table 4 simulation 1: 結果

	[1]	[2]	[3]		[3]	[1]	[2]
1	35	0	0	1	35	0	0
2	0	35	0	2	0	34	1
3	0	0	30	3	0	0	30

(a) 正解

(b) ソーティング後

Table 5 simulation 2: 結果

	[1]	[2]	[3]		[3]	[1]	[2]
1	35	0	0	1	33	1	1
2	0	35	0	2	11	23	1
3	0	0	30	3	0	0	30

(a) 正解

(b) ソーティング後

Table 6 simulation 3: 結果

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ソーティング前 (初期値)	2	1	2	1	1	2	3	2	2	1	1	3	2	1	3	3	2	1	2	2
ソーティング後	2	1	2	2	1	2	3	1	3	1	1	3	3	1	3	3	2	1	2	2

(a) 各スパイクのソーティング前後のラベル

	[1]	[2]	[3]		[2]	[3]	[1]		[2]	[3]	[1]
1	12	0	0	1	5	3	4	1	10	2	0
2	0	7	0	2	4	0	3	2	0	4	3
3	0	0	1	3	0	1	0	3	0	0	1

(b) 正解

(c) 初期値

(d) ソーティング後

Table 7 simulation 4: 結果

	[1]	[2]	[3]		[1]	[2]	[3]		[1]	[2]	[3]
1	35	0	0	1	33	1	1	1	29	1	5
2	0	35	0	2	11	23	1	2	7	22	6
3	0	0	30	3	0	0	30	3	11	17	3

(a) 正解

(b) 初期値

(c) ソーティング後

Table 8 simulation 5: 結果

	[1]	[2]	[3]
1	35	0	0
2	0	35	0
3	0	0	30

(a) 正解

	[1]	[2]	[3]
1	33	1	1
2	11	23	1
3	0	0	30

(b) 初期値

	[3]	[1]	[2]
1	31	3	1
2	15	19	1
3	0	0	30

(c) 91 回目

	[3]	[1]	[2]
1	31	3	1
2	12	22	1
3	0	0	30

(d) 92 回目

	[3]	[1]	[2]
1	31	3	1
2	15	19	1
3	0	0	30

(e) 93 回目

	[3]	[1]	[2]
1	31	3	1
2	14	20	1
3	0	0	30

(f) 94 回目

	[3]	[1]	[2]
1	31	3	1
2	6	28	1
3	0	0	30

(g) 95 回目

	[3]	[1]	[2]
1	31	3	1
2	14	20	1
3	0	0	30

(h) 96 回目

	[3]	[1]	[2]
1	31	3	1
2	9	25	1
3	0	0	30

(i) 97 回目

	[3]	[1]	[2]
1	31	3	1
2	1	33	1
3	0	0	30

(j) 98 回目

	[3]	[1]	[2]
1	31	3	1
2	15	19	1
3	0	0	30

(k) 99 回目

	[3]	[1]	[2]
1	31	3	1
2	6	28	1
3	0	0	30

(l) 100 回目

スパイクソーティングの結果を行列を用いて表した。行に正解のスパイクラベルを、列にソーティングの結果割り当てられたスパイクラベルをおいた。例えば、実際のスパイクラベルが1であるスパイクが、ソーティングの結果[2]とされれば、1行2列にカウントされる。各シミュレーションの結果について、左端から、仮に全問正解したときの行列、次に初期値のスパイクラベルの行列、ソーティング後の結果のスパイクラベルの行列をおいた。シミュレーション5については、各スパイクのスパイクラベルの選択を91回行った状態でのスパイクラベルの結果から、100回行った最終のスパイクラベルの結果までの行列をおいた。